

Connect:Direct[®] for z/OS

User's Guide

Version 5.0

**Connect:Direct for z/OS User's Guide
Version 5.0**

First Edition

(c) Copyright 1998-2010 Sterling Commerce, Inc. All rights reserved. Additional copyright information is located at the end of this document.

STERLING COMMERCE SOFTWARE

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

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

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

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

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

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

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

Sterling Commerce, Inc.

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

Contents

Chapter 1 About Connect:Direct for z/OS	9
Connect:Direct for z/OS Components	9
Connect:Direct for z/OS Configurations	10
Connect:Direct for z/OS Documentation	12
About This Guide	12
Task Overview	13
Chapter 2 Introduction to Connect:Direct Commands	15
About Connect:Direct Commands	15
Supported Commands	16
Writing Connect:Direct Commands	17
Native Format	17
Command Example	18
Command Syntax	18
Asterisks	18
Commas	19
Continuation Marks	19
Parentheses	19
Symbolic Substitution	19
Single and Double Quotation Marks	19
Comments	20
Concatenation	21
Special Characters	21
Special Purpose Bracketing	22
Indicating Case Sensitivity	22
Indicating Selection Criteria	23
Indicating Output Destination	24
Chapter 3 The Batch Interface	25
About the Batch Interface	25
Requirements	25
Processing Rules	26
Sample Job Stream to Run the Batch Interface	26
DDNAMES for DMBATCH	27
DMBATCH EXEC Parameters	27

DMBATCH Return Codes with DMCHLAPI	28
Synchronizing Submitted Processes	29
Using the MAXDELAY Keyword Parameter	29
Chapter 4 The Interactive User Interface (IUI)	31
The Primary Options Menu	32
Navigating the Interactive User Interface	34
Using Branch Commands	34
Using the PF Keys	35
Browsing the IUI Display	35
Displaying Your User Authorization	36
Getting Help	36
Operator Tables	37
Messages	38
Message Example	38
Messages in a Connect:Direct/Plex Environment	40
Viewing and Printing Messages Using Different Criteria	40
Viewing a Message List	42
Printing Messages	43
Chapter 5 The Application Program Interface	45
Communicating with the Connect:Direct API	45
The DMCHLAPI Program	46
Required Parameters	46
Extracting Return Codes	48
UICB Fields	49
DMCHLAPI Return Codes	53
Extract Fields for DMCHLAPI Return Codes	53
DMCHLAPI Examples	54
Sample Job Stream for Executing the Program	55
Chapter 6 Managing Sessions	57
Signing On to Connect:Direct for z/OS	57
Timeout Problems	57
Using SIGNON through the IUI	58
Using SIGNON through the Batch Interface	61
Signing On to the API	66
Signing On to Multiple Sessions	67
Using the Batch Interface for Multiple Session SIGNON	68
Using the IUI Multiple Session SIGNON Command	68
Swapping Nodes	69
Using SWAP NODE through the IUI	69
Using SWAP NODE through the Batch Interface	70
Signing Off of Connect:Direct	70
Using SIGNOFF through the IUI	71
Using SIGNOFF through the Batch Interface	71

Sequencing the SIGNON, SWAP NODE, and SIGNOFF Commands	72
Example 1: Terminating Two Sessions with One SIGNOFF Command	72
Example 2: Terminating a Master Session with the SIGNOFF Command	72
Example 3: Terminating a Non-Master Session with the SIGNOFF Command	73

Chapter 7 Building, Modifying, and Submitting Processes **75**

Overview of Process Execution	75
Process Routing	76
Process Queuing	77
Process Selection	77
Intelligent Session Retry	78
Process Execution Example	79
Process Management in a Connect:Direct/Plex	81
Example	82
The SUBMIT Command	84
Command Examples	93
Process Example	93
Symbolic Substitution Example	93
Submitting Processes through the Batch Interface	93
Submitting Processes through the IUI	94
IUI Submit Options	94
Validating Processes from the SB, DF, and CF IUI Options	95
Using the SB IUI Option to Submit a Predefined Process	95
Using the DF IUI Option to Create, Edit, and Submit Processes	98
Modifying and Submitting a Process	98
Creating and Submitting a New Process	99
Submitting a Process	100
Using Statement Models to Create a Process	100
Using the CF IUI Option to Generate a COPY Process	101

Chapter 8 Controlling Processes in the TCQ **109**

Controlling Processes with Commands	109
Setting Selection Criteria	110
Modifying a Process in the TCQ with CHANGE PROCESS	110
Command Format	111
Command Examples	114
Issuing CHANGE PROCESS through the Batch Interface	115
Issuing CHANGE PROCESS through the IUI	115
Suspending, Flushing, and Deleting Processes	116
SUSPEND, FLUSH, or DELETE PROCESS Command Format	117
Command Examples	118
SUSPEND Process	118
DELETE Process	118
FLUSH Process	118
Issuing the Commands through the Batch Interface	119
Issuing the Commands through the IUI	119
Examining Processes in the TCQ	121
SELECT PROCESS Command Format	121
SELECT PROCESS Command Example	124
Issuing SELECT PROCESS through the Batch Interface	124
Issuing SELECT PROCESS through the IUI	124

Chapter 9 Process Queuing and Recovery 131

The Transmission Control Queue (TCQ)	131
Logical Queues	131
Queuing Parameters	132
TCQ Status and State Values	132
Wait Queue	133
Execution Queue	134
Hold Queue	138
Process Retention Queue	139
Timer Queue	139
Process Recovery	140
COPY Statement Checkpoint/Restart Facility	141
Initialization Parameters	141
Checkpoint/Restart File	142
COPY Statement Parameters	142
Checkpoint/Restart Examples for TCP/IP, UDT, or LU6.2 Transfers	143
TCP/IP, UDT, or LU6.2 Transfers	143
Other Transfers	143
RUN TASK Checkpoint/Restart	144
Checkpoint Records	144
Restart Procedure	144
Re-execution of the RUN TASK	145

Chapter 10 Process Results and Statistics 147

Statistics Log Records	147
Statistics Records Content	148
Elapsed Time Significance	149
InterConnect Option Records	149
SELECT STATISTICS Command	150
Command Format	150
Using SELECT STATISTICS through the Batch Interface	156
Command Examples	156
Sample Output	158
Using SELECT STATISTICS through the IUI	159
Statistics Summary	163
Display Statistics	165
SELECT MESSAGE Command	166

Chapter 11 The Network Map 169

Retrieving Records from the Network Map File	169
Issuing the SELECT NETMAP Command from the IUI	169
Issuing the SELECT NETMAP Command from the Batch Interface	171
SELECT NETMAP Command Format	172
Translating TCP/IP Host Names to Network Addresses	173
Issuing the SELECT TCPXLAT Command from the IUI	173
Issuing the SELECT TCPXLAT Command with the Batch Interface	174
SELECT TCPXLAT Command Format	175

Chapter 12 Utility Programs **177**

User Notification Programs	177
Operator Console Notify Program (DMNOTIFY)	178
User Notify Program (DMNOTFY2)	179
Dynamic Allocation Program (DMRTDYN)	180
Parameters	180
Sample Program Uses	181
Run Task Wait Program (DMRTWAIT)	182
VSAM AMS Interface Program (DMRTAMS)	183
Parameters	184
Sample Program	184
Symbolic Resolution Utilities (DMRTSUB and RTGDGSUB)	184
Passing Variables Using RTGDGSUB	186
Using Symbolic Variables	187
Connect:Direct Exits and DMRTSUB	187
Error Output	188
Return Codes	188
Batch Compression Utility (CDSACOMP)	188
Parameters	189
Examples	189
Output	190
Testing the Effects of Changing Values for Extended Compression Parameters	190
Automatic Decompression	191
Determine 'High-Water Mark' for a Period (DMRTVITL)	192
ADRDSIU Interface Program (DMDSSIOX I/O Exit)	193
Sample Processes	193
Running Traces for the DMDSSIOX I/O Exit	193
Limitations	193
Connect:Direct FTP+ for z/OS (CDFTP)	194
Components of Connect:Direct FTP+	194
Security Considerations	194
Configure Connect:Direct FTP+ for z/OS	195
Install and Configure Connect:Direct FTP+ Plug-in	195
Create CDFTP Procedure	195
Create JCL to Start the Connect:Direct FTP+ Manager	197
Update Netmap for Remote Nodes	198
Start Connect:Direct FTP+ Manager	199
Modify FTP JCL	199
New PARMLIB and SAMPLIB Members	201
Supported FTP Commands	201
Fully Supported FTP Commands	202
Partially Supported FTP Commands	203
FTP Commands Accepted, But Ignored	206

Glossary **207**

Index **227**

About Connect:Direct for z/OS

Connect:Direct is peer-to-peer file-based integration middleware optimized for assured delivery, high-volume and secure data exchange within and between enterprises. It is optimized for high performance and moves files containing any type of data across multiple platforms, disparate file systems, and disparate media.

Connect:Direct enables businesses to:

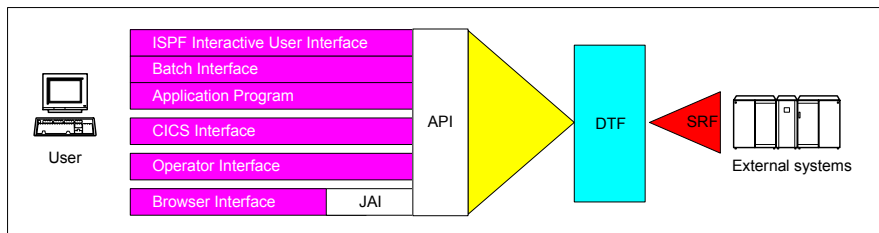
- ◆ Move large amounts of data internally and externally
- ◆ Share information with business partners
- ◆ Schedule business information-related application activities
- ◆ Automate data distribution
- ◆ Control and audit network activities
- ◆ Maintain network security
- ◆ Use one common command structure for information management with platform-specific interfaces

Connect:Direct for z/OS Components

The major components of Connect:Direct for z/OS are:

- ◆ Data Transmission Facility (DTF), which executes user commands and Processes
- ◆ Application Program Interface (API), which enables user interfaces to communicate with the DTF
- ◆ User interfaces, such as the Interactive User Interface (IUI), browser end user interface (EUI), the batch interface, and the CICS interface, which interacts between the user and the API.

The following illustration shows these components.



Connect:Direct for z/OS Configurations

Connect:Direct for z/OS can be used in either of the following configurations:

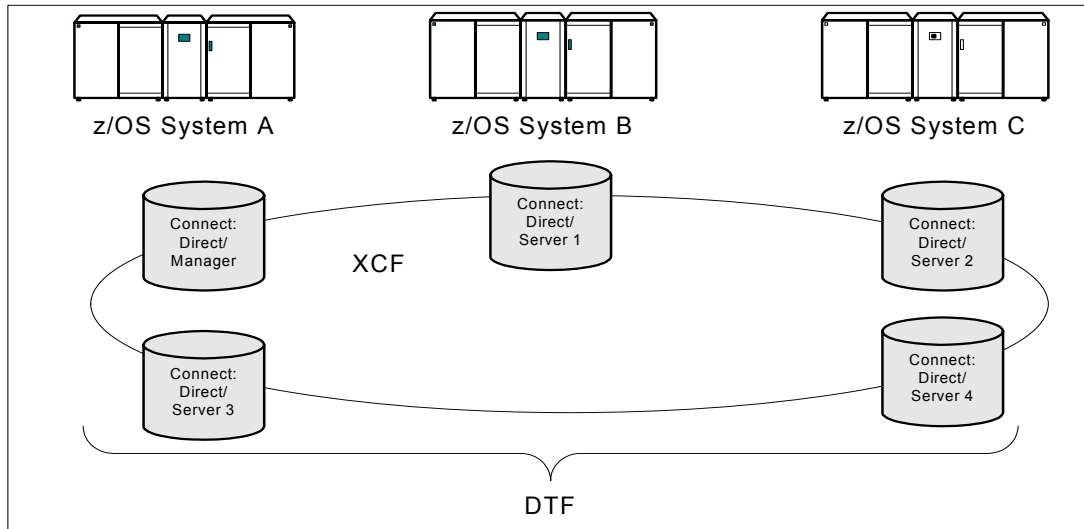
- ◆ Connect:Direct Stand-alone Server is a stand-alone Connect:Direct system operating in an IBM z/OS environment.
- ◆ Connect:Direct/Plex is a Connect:Direct system operating in an IBM z/OS sysplex or parallel sysplex environment. A Connect:Direct/Plex consists of:
 - ◆ Connect:Direct Manager, which manages the Connect:Direct/Plex. This includes handling:
 - Workload distribution and balancing
 - Processing queues
 - IUI and API components
 - System-wide parameters
 - Network maps (netmaps)
 - Files that control user authorization, file attributes, and statistics logging
 - Secure+ parameters
 - ◆ One or more Connect:Direct Servers, which execute Processes assigned by the Connect:Direct Manager. Each Connect:Direct Server can be on a different z/OS system, or there can be multiple servers in one system. Individual servers can independently join or leave the Connect:Direct/Plex.

The Connect:Direct Manager and all Connect:Direct Servers form the DTF in a Connect:Direct/Plex environment.

A Process runs on any Connect:Direct Server in a Connect:Direct/Plex environment. You can balance workload dynamically among the Connect:Direct Servers. You can also direct

Processes to specific servers (by specifying the PLEXCLASS keyword in the PROCESS statement). For example, a Process that requires a tape drive can be directed to a Connect:Direct Server running on a computer with tape drives.

The Connect:Direct Manager communicates with the Connect:Direct Servers through the IBM z/OS Cross-System Coupling Facility (XCF), as in the following illustration.



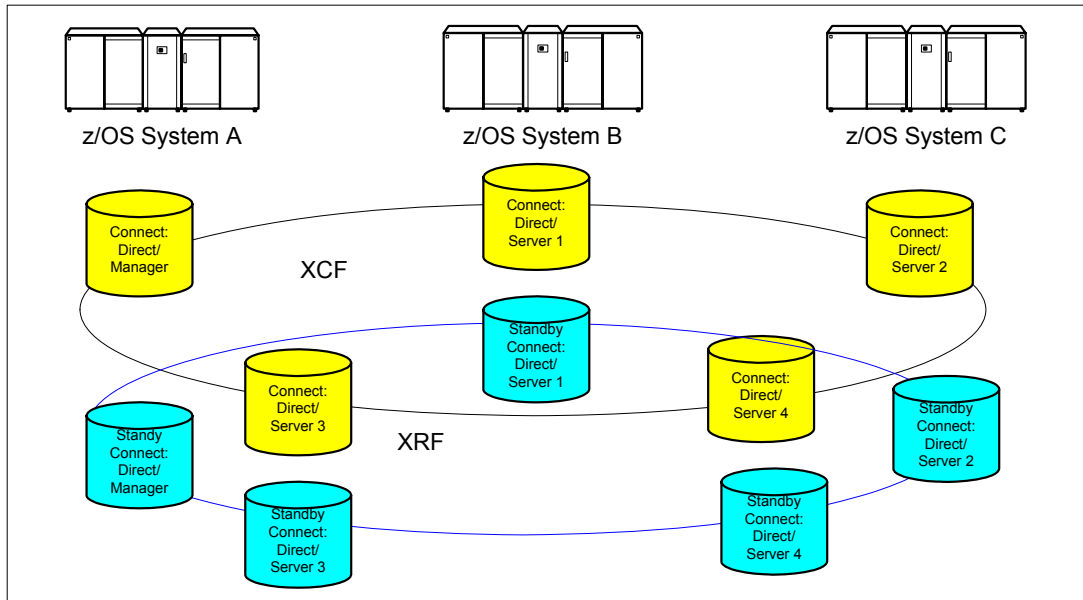
Connect:Direct/Plex operation is transparent; it appears as a single local node to other systems that communicate with it.

Note: For more information about XCF, see the IBM parallel sysplex documentation library. No additional XCF setup is required to use Connect:Direct/Plex.

Connect:Direct runs as a single system image in either the Connect:Direct Stand-alone Server or Connect:Direct/Plex environments. It starts as a single job. The Connect:Direct system shares parameters, the network map, and processing queues throughout the Connect:Direct system.

Both Connect:Direct Stand-alone Server and Connect:Direct/Plex can use the z/OS Extended Recovery Facility (XRF) to support extended recovery. If you use extended recovery in a Connect:Direct Stand-alone Server, a standby Connect:Direct system waits to assume the work of a failing system.

If you use extended recovery in a Connect:Direct/Plex environment, each active member has a standby member ready to assume work if the active member fails, as shown in the following illustration.



If you are using the dynamic VIPA resources, the standby managers or servers do not have to reside in the same z/OS system as the active manager or server they are backing up.

If you are not using the dynamic VIPA resources, the standby managers or servers must reside in the same z/OS system as the active manager or server they are backing up.

Connect:Direct for z/OS Documentation

See *Connect:Direct for z/OS Release Notes* for a complete list of the product documentation.

About This Guide

Connect:Direct for z/OS User's Guide is for programmers and network operations staff who use Connect:Direct for z/OS.

This guide assumes knowledge of the IBM z/OS operating system, Job Control Language (JCL), and Interactive Systems Productivity Facility (ISPF). If you are not familiar with the z/OS operating system, refer to the z/OS library of manuals.

Read the first two chapters of the book to gain general knowledge of Connect:Direct. These chapters introduce you to the basic components and general concepts, and summarize the Connect:Direct commands. The next three chapters focus on the different interfaces you can use with Connect:Direct including the Batch Interface, Interactive Interface, and Application Program Interface. The remaining chapters deal with specific tasks you can perform using Connect:Direct.

Task Overview

The following table directs you to the information required to perform Connect:Direct tasks:

Task	For More Information, See
Signing on and off Connect:Direct and swapping nodes	Chapter 6, <i>Managing Sessions</i>
Submitting Processes and learning command syntax using both the IUI and Batch Interface	Chapter 7, <i>Building, Modifying, and Submitting Processes</i>
Controlling and monitoring Processes after they are submitted to the Transmission Control Queue (TCQ)	Chapter 8, <i>Controlling Processes in the TCQ</i>
Accessing queues in the TCQ, and manipulating and recovering Processes in the TCQ	Chapter 9, <i>Process Queuing and Recovery</i>
Retrieving statistics records and monitoring the results of Processes after they have been executed	Chapter 10, <i>Process Results and Statistics</i>
Retrieving records from the network map file and translating TCP/IP host names to network addresses	Chapter 11, <i>The Network Map</i>
Using Connect:Direct utility programs and I/O exits to facilitate your use of the software, such as dynamically allocating data sets and compressing and decompressing files stored in a ZLIB compressed format	Chapter 12, <i>Utility Programs</i>

Introduction to Connect:Direct Commands

This chapter provides an overview of Connect:Direct commands and describes how to write Connect:Direct commands.

About Connect:Direct Commands

You use Connect:Direct commands to submit and manipulate Processes in the Transmission Control Queue (TCQ), where the Processes are stored. For example, you can select, delete, and suspend Processes using Connect:Direct commands.

Other commands allow you to perform administrative functions such as creating and updating network maps, user authorities, and Type files. This guide describes the following two methods of issuing commands:

Method	Description
Batch Interface	Enables you to issue Connect:Direct commands with the DMBATCH program in a batch environment. Review Chapter 3, <i>The Batch Interface</i> , for more information about how to use the Batch Interface.
Interactive User	Provides an easy-to-use method of command entry through fill-in-the-blank screens. The Interactive User Interface (IUI) is an ISPF Dialog online interface. Review Chapter 4, <i>The Interactive User Interface (IUI)</i> , for more information on how to use the IUI.

Note: Some command options which you can select through the IUI do not have a batch equivalent. The documentation notes these exceptions in option descriptions.

Supported Commands

The following table lists the Connect:Direct commands, the IUI command abbreviations, and their function.

Command	IUI	Function
CHANGE PROCESS	CP	Modify a Process in the TCQ
DELETE PROCESS	DP	Remove a Process from the TCQ
DELETE TYPE ²	DT	Delete a record from the Type file
DELETE USER ¹	DU	Delete a user from the Authorization file
FLUSH PROCESS ²	FP	Terminate an Executing Process
FLUSH TASK ¹	TF	Remove a task from the Execution queue
INQUIRE ¹	INQ	View information about Connect:Direct DTF status
INQUIRE SNMP	INQ SNMP	Display SNMP Trap Table
INSERT TYPE ²	IT	Insert a record in the Type file
INSERT USER ¹	IU	Add a user to the Authorization file
MODIFY ¹	MD	Run Connect:Direct diagnostics and modify initialization parameters.
SELECT MESSAGE	SEL MSG	Display message detail using message ID
SELECT NETMAP	NM	Display node from the network map file
SELECT PROCESS	SP	Examine a Process in the TCQ
SELECT STATISTICS	SS	Examine records in the Statistics Log
SELECT TASK ¹	TS	Display Connect:Direct system tasks status
SELECT TCPXLAT ¹	NM	Translate TCP/IP host names - addresses
SELECT TYPE ²	ST	Examine the records in the Type file
SELECT USER ¹	SU	Examine user authorization
SIGNON		Connect to Connect:Direct DTF
SIGNOFF	X	Terminate a connection to the Connect:Direct DTF
STATISTICS ¹	STAT	Perform statistics functions
STOP CD ¹	SN	Stop Connect:Direct operation

1 Administrative command. Refer to *Connect:Direct for z/OS Administration Guide* for more information on administrative commands.

2 Both a user and administrative task. Refer to *Connect:Direct for z/OS Administration Guide* for more information.

Command	IUI	Function
SUBMIT	SB	Submit a Process for execution
SUSPEND PROCESS	SUS	Suspend Process execution
SWAP NODE	SW	Swap to another Connect:Direct node
UPDATE NETMAP ¹	UNM	Update the network map file
UPDATE TYPE ²	IT	Update a record in the Type file
UPDATE USER ¹	IU	Change user privileges

- 1 Administrative command. Refer to *Connect:Direct for z/OS Administration Guide* for more information on administrative commands.
- 2 Both a user and administrative task. Refer to *Connect:Direct for z/OS Administration Guide* for more information.

Writing Connect:Direct Commands

A Connect:Direct command is a string of characters that conveys your requests to the Connect:Direct DTF for execution. You write these commands using a unique syntax called the Connect:Direct native format.

If you are using the batch interface, operator interface, or your own user-written application, you write the commands in this native format. If you are using the IUI, Connect:Direct builds the commands in the format for you.

Native Format

The following figure shows the structure of a Connect:Direct command in its native format.

Label Command Parameters and Subparameters

The following table explains each command component:

Component	Description
Label (optional)	You can optionally identify your Connect:Direct command through use of a label. A label must begin in position 1 and consists of a 1-8 character alphanumeric string, with the first character alphabetic.
Command	The command itself specifies the Connect:Direct function requested and must begin after position 1. Use one or more blanks to separate the command from the parameters which follow it.

Component	Description
Parameters or Subparameters	<p>Parameters or subparameters specify further instructions for the command. Separate your parameters by one or more blanks. Parameters can be either keyword or positional.</p> <p>Keyword parameters are usually followed by an equal sign and can have a set of subparameters. An example of a keyword parameter is CASE in the following command.</p> <pre>SIGNON NETMAP=network.map.name CASE=Yes No</pre> <p>You must type positional parameters in a specific order, with commas replacing any parameter not typed. These parameters are always on the right of the equal sign. Enclose positional subparameters in parentheses, with the parentheses preceded and followed by blanks or commas.</p> <p>In the following command USERID is an example of a keyword parameter, and ID,pswd,newpswd are examples of positional subparameters.</p> <pre>SIGNON USERID=(ID,pswd,newpswd)</pre> <p>A positional parameter or the variable information in a keyword parameter is sometimes a list of subparameters. Such a list can include both positional and keyword parameters.</p>

Command Example

The following CHANGE PROCESS command changes a preexisting Process named PAYROLL so that the new destination node is DALLAS1 and the Process executes every Friday.

```
CH PROC WHERE (PNAME=PAYROLL) -
        DEST=DALLAS1 -
        RETAIN=Y -
        STARTT= (FR)
```

For instructions on creating a Process, see the [Connect:Direct Processes Web site](#).

Command Syntax

This section describes the syntax that you use to construct Connect:Direct commands and Processes. For a complete description of Process syntax and examples, see the [Connect:Direct Processes Web site](#).

Asterisks

Use asterisks to indicate generic specifications of parameters in the SELECT commands. With generics, you request information by specifying just a single asterisk (*) or a character string plus an asterisk.

To examine records for users whose user IDs begin with ST, specify the following parameter.

```
USERID=ST*
```

Commas

Commas have two functions: (1) to separate items within a list (except in the case of symbolic substitution. Refer to *Symbolic Substitution* on page 19), and (2) to control the order of values specified as positional parameters. You must use a comma to indicate omission of a positional parameter. In the following example, the omission of the `pswd` subparameter is indicated by the extra comma.

```
SIGNON USERID=(ID, ,newpswd)
```

Continuation Marks

A command can continue across multiple lines. Use the hyphen (-), preceded and followed by a space, to indicate that the command continues on the following line. The hyphen can appear anywhere in positions 3-79. The following command is continued on a second line.

```
CHANGE PROCESS WHERE (PNAME=PAYROLL) -  
      DEST=DALLAS1
```

Parentheses

In the following example, parentheses enclose lists and associate a group of values.

```
SIGNON USERID=(MYUID1,MYPSWD)
```

Symbolic Substitution

Use symbolic substitution to substitute information in a Connect:Direct Process. The substitution is represented by an ampersand (&) plus 1-8 alphanumeric characters. In the following example, the value for `&DATA` is resolved when you submit the Process.

```
DSN=&DATA
```

If you have multiple symbolics, separate them with one or more spaces.

Single and Double Quotation Marks

The rules for using single and double quotation marks are:

- ◆ Single-quote strings allow the parsing of parameters as typed.
- ◆ Double-quote strings allow the resolution of symbolic substitution in a quoted string.

The following example shows the use of single quotation marks to enable you to embed special characters or blanks within a parameter or subparameter value.

```
SIGNON PACCT='JOB FOR SYSMAINT,DEPT.27'
```

The following example shows the use of double quotation marks to allow for the resolution of symbolic substitution.

```
SIGNON PACCT="JOB FOR SYSMOINT,DEPT. &DEPND"
```

Single quotation marks are not valid for symbolic substitution in a Connect:Direct command but can be used with a keyword parameter in a Process statement (refer to *Process Language Syntax* on the [Connect:Direct Processes Web site](#)). Double quotation marks are dropped from a string before a symbolic substitution is resolved while single quotation marks are not dropped and result in an invalid Process.

For example, the following SUBMIT Process command, which uses single quotation marks:

```
SUB PROC=DCB HOLD=YES -
  &DCB='DCB=(LRECL=80,BLKSIZE=3120,RECFM=FM)'
```

results in the following invalid Process.

```
SUB PROC=DCB HOLD=YES      &DCB='DCB=(LRECL=80,BLKSIZE=3120,RECFM=FB)'
```

The following example which uses the same Process information as the example above but this time with double quotation marks:

```
SUB PROC=DCB HOLD=YES -
  &DCB="DCB=(LRECL=80,BLKSIZE=3120,RECFM=FM)"
```

resolves to:

```
SUB PROC=DCB HOLD=YES      &DCB=DCB=(LRECL=80,BLKSIZE=3120,RECFM=FB)
```

Comments

Comments allow you to include descriptive information within a command. Comments are available for your convenience and do not affect Connect:Direct. Use the following formats for comments:

- ◆ Preceded by a slash-asterisk (/*) and followed by an asterisk-slash (*/).
- ◆ An asterisk (*) in position 1, followed by the comment.

All three of the following lines are valid comments, indicating that the SIGNON command labeled DSIGN signs on to the DALLAS node.

```
/* SIGN ON TO DALLAS */
*SIGN ON TO DALLAS
DSIGN SIGNON NODE=DALLAS /*DALLAS*/
```

Concatenation

Use the double bar (||) to concatenate, or link together, character strings. You must precede the double bar and follow it with a blank. Use the double bar to join a long value that continues over multiple records. For example, you can type the following command.

```
PACCT=JOBACCTDATA || -  
WITHNOBLANKS
```

Note: The vertical bar is x'4F'.

Connect:Direct resolves your command to the following format.

```
PACCT=JOBACCTDATAWITHNOBLANKS
```

Special Characters

Certain characters cause Connect:Direct to take special actions. These special characters are the hyphen (-), double bar (||), ampersand (&), and the Connect:Direct delimiters. The following table lists the delimiters recognized by Connect:Direct.

Delimiters	Description
	blank
<	less than sign
>	greater than sign
*	asterisk
(open parenthesis
)	close parenthesis
¬	not sign
/	slash
\	backslash
,	comma
.	period
'	single quotation mark
"	double quotation mark
=	equal sign
{	opening brace

Delimiters	Description
}	closing brace
[opening bracket
]	closing bracket

Special Purpose Bracketing

You must often maintain special characters as part of a string. To maintain special characters, enclose the string in bracketing characters. Bracketing characters are backslashes (\), single quotation marks ('), and double quotation marks (").

Bracketing backslashes are indicators of special processing of a character string. Connect:Direct does not maintain them as part of the string at its final resolution. Use bracketing backslashes to:

- ◆ Continue a string containing special characters across multiple lines
- ◆ Ensure that quotation marks within the string are maintained

The following is an example of using bracketing backslashes in a command.

```
PACCT=\ 'DEPT\MIS\ || -
        \602'\
```

Connect:Direct resolves the command as follows.

```
PACCT='DEPT\MIS602'
```

Indicating Case Sensitivity

The **CASE** parameter specifies whether parameters associated with accounting data, user ID, password, and data set name in the commands and Processes are case sensitive. You can make this designation at the signon level for all commands issued for the session established by the signon, and at the command level. The following table describes the levels at which you can set the CASE parameter.

Level	Description
Session	The CASE parameter of the SIGNON command indicates if commands entered during this session are case sensitive. The default value is NO. You can override this setting at the command level.

Level	Description
Individual Command	<p>You can change the CASE parameter for an individual command and override the session case sensitivity as follows:</p> <ul style="list-style-type: none"> ◆ In the Connect:Direct for z/OS IUI, the CASE parameter is usually indicated on the IUI by the following words: DO YOU WANT VALUES FOR THIS REQUEST TO BE CASE SENSITIVE? ==> Reply Y or N to this question. The value for the CASE parameter stays in effect as long as you reuse the same panel. When you change panels, the value reverts to the session default. ◆ In native command mode, such as used in the Batch Interface, you must designate your override on each command record if you want to override the CASE parameter of the SIGNON command.
Process	<p>For commands that refer to Processes already in the TCQ, the CASE parameter only applies to the fields in the command. For commands that refer to Processes that are not yet in the TCQ, the case sensitivity applies to the Process statements.</p>

Indicating Selection Criteria

Use the **WHERE** parameter to specify which records to select, change, or delete based on subparameters. You can specify subparameter values such as Process name or Userid. The selection subparameters are optional; however, you must specify at least one. These subparameters are special in that they identify which records are selected for the command activity defined by the other parameters.

Use the WHERE parameter to indicate selection criteria for commands which affect Process, Task, Type, User, and Statistic records. The command function applies to all records which match the selection criteria.

For example, in the CHANGE PROCESS command, you can use the WHERE parameter with its PNUMBER subparameter to select a list of Processes by number and release all of them through the RELEASE parameter.

In the IUI, the WHERE parameter is represented by the list of selection criteria which precedes the command parameters themselves. For example, you see the following on the CHANGE PROCESS screen.

```

PROCESS NUMBERS :
==> _____ ==> _____ ==> _____ ==> _____

```

Additionally, you see lists for Process name and submitter to indicate the selection criteria. The Connect:Direct IUI builds the WHERE parameter for the command as it selects the fields from left to right on the screen.

Indicating Output Destination

The **FILE** | **PRINT** | **TABLE** | **DISPLAY** parameters specify the form in which information is presented when the command produces output. The following table describes each parameter.

Parameter	Description
FILE	<p>Causes Connect:Direct to return the output as unformatted records to the temporary file. You indicate the name of your temporary file through the TMPDSN parameter of your SIGNON command. See Chapter 6, <i>Managing Sessions</i>, for a description of the TMPDSN parameter. The statistics record macros in the Connect:Direct sample library provide the output record format. These macros are listed in the <i>Connect:Direct for z/OS Administration Guide</i>.</p> <p>When Connect:Direct puts the unformatted records in a temporary file, other programs can process the records and customize reports for specific needs. You can use this functionality to process statistics records. You can archive a statistics file each day by means of a Generation Data Group (GDG). The Connect:Direct product provides sample job streams that extract statistics on a daily basis.</p>
PRINT	<p>Routes output of the command to the destination specified in the PRINT keyword of the SIGNON command. See Chapter 6, <i>Managing Sessions</i>, for a description of the PRINT parameter.</p>
TABLE or DISPLAY	<p>Stores the output of the command in the Connect:Direct temporary file or file specified in the TMPDSN parameter and displays it upon successful completion of the command. (For more information on the TMPDSN parameter, see page 61.) The output is usually in tabular format. You can browse the output through the use of such commands as UP, RIGHT, or FIND. Press the PF1 key for Help on how to browse the file.</p>

Some screens offer additional output format, such as summary tables. These options are listed along with the display, file, and print options.

The Batch Interface

This chapter provides information about using the Batch Interface, and serves as an introduction to the more task-specific chapters which follow.

About the Batch Interface

You request Connect:Direct services in the batch environment through the Batch Interface program, DMBATCH. When you submit a job stream that contains the Connect:Direct commands as SYSIN input, DMBATCH reads the input data stream and processes the requested functions. DMBATCH supports all Connect:Direct commands.

For the command structure of the Connect:Direct commands you can issue through the Batch Interface, refer to the *Connect:Direct for z/OS Administration Guide* for administrative commands, and this manual for user commands.

Requirements

Your Batch Interface job must meet the following requirements:

- ◆ You cannot use Process statements in the job stream. Use the SUBMIT command with a preexisting Process that contains Process statements.

Note: Unless MAXDELAY is used, DMBATCH processing of the SUBMIT is asynchronous; the batch job is notified of a successful SUBMIT, but not whether the PROCESS itself is successful.

- ◆ The Processes you submit reside in the Connect:Direct Public Process Library, allocated to the DMPUBLIB DD statement in the DMBATCH JCL. If the Process does not reside in the Connect:Direct Public Process Library, use the DSN parameter of the SUBMIT command to indicate the location of the Process. See Chapter 7, *Building, Modifying, and Submitting Processes*, for the DSN parameter description.
- ◆ The Connect:Direct commands you use in the batch job stream must follow the syntax outlined in *Writing Connect:Direct Commands* on page 17.

- ◆ If you are not using the Extended Submit Facility (ESF), the DTFs that you sign on to must be active when you submit the DMBATCH job.
- ◆ You can specify ESF as a SIGNON command parameter. You can only issue SIGNON, SIGNOFF, and SUBMIT commands using ESF. ESF is only available for Processes you submit on the local node.

Processing Rules

The following rules apply when you type Connect:Direct commands and options in the DMBATCH job stream:

- ◆ The first command in the job stream must be a SIGNON command.
- ◆ The maximum command string length is 4,096 bytes. Each new Connect:Direct command in the job stream must start on a separate line. You can split a command across more than one line using the continuation mark, but the first word or string of the next command must start on a new line. A sample follows.

```
SIGNON USERID=(LYNN) -
        NODE=CD.BOSTON
SEL STAT WHERE (PNUM=24)
SIGNOFF
```

Sample Job Stream to Run the Batch Interface

The following figure shows a sample job stream to run the DMBATCH program. The job stream is located in \$CD.JCL(DMBATCH).

```
//JOBNAME JOB (ACCNTNG), PROGRAMMER, TIME=2, CLASS=A,
//          MSGCLASS=A
//DMBATCH EXEC PGM=DMBATCH, REGION=1024K, PARM='YYSLYNN'
//STEPLIB DD DISP=SHR, DSN=$CD.LINKLIB
//DMNETMAP DD DISP=SHR, DSN=$CD.NETMAP
//DMPUBLIB DD DISP=SHR, DSN=$CD.PROCESS
//DMMSGFIL DD DISP=SHR, DSN=$CD.MSG
//DMPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN DD *
SIGNON USERID=(USERID, PASSWORD)
SUBMIT PROC=COPY SNODE=CD.NODE.B -
        HOLD=NO -
        &NODE=PNODE -
        &DSN1=DATASET1 -
        &DSN2=DATASET2 -
        &DISP1=RPL -
        &DISP2=CATLG -
        &COMPRESS=COMPRESS
SEL PROC WHERE (QUEUE=A) TABLE
SIGNOFF
/*
```

The following figure shows the COPY Process submitted in the previous example as it is displayed in the Connect:Direct Public Process Library, \$CD.PROCESS.

```

COPY      PROCESS
STEP1    COPY      FROM (&NODE DSN=&DSN1  DISP=SHR)    -
          TO       (DSN=&DSN2  DISP= (&DISP1, &DISP2)) -
          &COMPRESS

```

DDNAMES for DMBATCH

The following table describes the required and optional ddnames for the DMBATCH program shown in the example, *Sample Job Stream to Run the Batch Interface* on page 26:

DDNAME	Description
STEPLIB	Indicates the location of the library containing the Connect:Direct load modules. This DD statement is required, unless the data set is in linklist.
DMNETMAP	Indicates the name of the network map for the node you are signing on to. You can also specify this name through the SIGNON command NETMAP parameter. If you use both methods, the network map file name specified on the SIGNON command takes precedence. This DD statement is optional.
DMPUBLIB	Indicates the library containing the Connect:Direct Processes. This DD statement is required.
DMMSGFIL	Indicates the Connect:Direct message file that contains Connect:Direct messages. This DD statement is required.
DMPRINT	Indicates the destination of the job output from DMBATCH that collects output messages such as error messages. This DD statement is required. Always check DMPRINT output for verification of the processing that has just occurred.
SYSPRINT	Indicates the destination of the job execution messages. This DD statement is optional.
NDMCMDS	Use to test new job streams. If specified, it prints an image of each command sent to Connect:Direct after processing all symbolic substitution and parameter overrides. This DD statement is optional.
SYSIN	Indicates the location of the Connect:Direct commands. You can represent it as a sequential file, PDS member, or instream data. This DD statement is required.

DMBATCH EXEC Parameters

The characters in the PARM keyword in the DMBATCH EXEC statement are required parameters.

These values identify the optional output formatting routines. Y stands for yes, N stands for no, S stands for short text, and L stands for long text. These fields are positional. No default values are used. The output is displayed in the data set defined by DMPRINT.

The following table describes the DMBATCH EXEC parameters.

Position	Value	Description
1	Y	Display the command string that executed.
	N	Do not display the command string that executed.
2	Y	Display the API return code and message ID.
	N	Do not display the API return code and message ID.
3	S	Display short message text when the API sends a return code of zero.
	L	Display long message text when the API sends a return code of zero.
	N	Do not display message text when the API sends a return code of zero.
4	S	Display short message text when the API sends a non-zero return code.
	L	Display long message text when the API sends a non-zero return code.
	N	Do not display message text when the API sends a return code greater than zero.
5	Y	Display the data that generated in the temporary file.
	N	Do not display the data that generated in the temporary file.
6	N	Reserved
7	N	Reserved
8	Y	Prints output from a SELECT STATS command to DMPRINT when a MAXDELAY Process fails with a return code other than 52.

DMBATCH Return Codes with DMCHLAPI

DMBATCH communicates with the Connect:Direct API through a high-level application interface program, DMCHLAPI. The DMCHLAPI program detects error situations while processing commands for DMBATCH, and issues special return codes based on these situations.

If you receive a return code of 4 or 8, always check the DMPRINT DD output for the exact error encountered during command processing to DMBATCH.

A return code of 8 normally indicates an invalid parameter or keyword. A return code of 4 indicates that DMCHLAPI found nothing for the command. For example, you could receive a return code of 4 if you issued a SELECT STATISTICS command for a Process with no statistics.

Note: The return code from DMBATCH reflects the highest return code of all commands processed.

If you receive a DMBATCH return code greater than 8, check the DMBATCH return codes in the following list for error information.

Return Code	Description
0000000C	A session is lost in a multiple session environment.
00000010	The master session is lost.
00000014	The master session is signed off successfully.
0000001C	A non master signon failed.
00000020	The high-level interface program, DMCHLAPI, received an invalid number of input parameters.
00000024	The output specification included an invalid parameter.
00000028	Invalid pointer to the UICB is passed to DMCHLAPI.

Refer to *Sample Job Stream to Run the Batch Interface* on page 26 for more information on multiple session environments.

Synchronizing Submitted Processes

This section describes how you can synchronize the operation of batch job streams and the execution of a Process with the MAXDELAY keyword parameter in a SUBMIT command or PROCESS statement.

Using the MAXDELAY Keyword Parameter

You can synchronize submitted Processes by coding the MAXDELAY keyword parameter in either the SUBMIT command or the PROCESS statement. The MAXDELAY keyword enables you to suspend the execution of the DMBATCH jobstep until the submitted Process either completes or a specified interval of time elapses.

For a detailed description of the MAXDELAY parameter, refer to *The SUBMIT Command* on page 84, or the [Connect:Direct Processes Web site](#) (and then go to the MAXDELAY parameter link from the PROCESS statement).

-
- Note:** Because of the following restrictions, you may want to code MAXDELAY in the SUBMIT command instead of the PROCESS statement:
- ◆ If you code MAXDELAY in a PROCESS and submit it through the IUI, your TSO session hangs until the Process completes.
 - ◆ The person performing the SUBMIT command may not be able to see or know about the MAXDELAY value specified in the PROCESS statement and wonder what is holding up the session.
-

To place a job stream in a wait status for a specified period of time and set return codes indicating whether the step or task completed successfully or not, you can use the DMRTWAIT program as a batch step or a RUN TASK. For more information, see *Run Task Wait Program (DMRTWAIT)* on page 182.

The following example shows the MAXDELAY keyword coded in the SUBMIT command.

```
//SYSIN      DD  *
SIGNON USERID=(USER01,PASSWRD)
SUBMIT PROC=PROCAAA MAXDELAY=UNLIMITED
SIGNOFF
/*
```

You can also place the MAXDELAY keyword in the PROCESS statement for PROCAAA, as in the following example.

```
PROCAAA  PROC  SNODE=REMOTE.NODE.B MAXDELAY=UNLIMITED
STEP01  COPY TO  (DSN=USER01.TESTDATA.OUTPUT  -
                DISP=RPL                      -
                SNODE)                        -
        FROM (DSN=USER01.TESTDATA.INPUT      -
                DISP=SHR                      -
                PNODE)
```

The Interactive User Interface (IUI)

This chapter provides information about using the Interactive User Interface (IUI), and serves as an introduction to the more task-specific chapters which follow.

You issue Connect:Direct commands using the IUI through two menus:

Menu	Description
The Primary Options Menu	Enables you to invoke screens where you can create and submit Processes, view statistics about your Processes, control your Processes in the Transmission Control Queue (TCQ), view your Connect:Direct environment, access the message file, sign on, swap nodes, and sign off Connect:Direct. This option is available to all users, and is described in this book.
The Administrative Options Menu	Enables the Connect:Direct product administrator to perform maintenance to the Type file, the Authorization file, and the network map and to perform functions such as selecting and flushing tasks, initializing traces, typing native commands, displaying storage utilization, requesting traces, and stopping Connect:Direct. These options are described in the <i>Connect:Direct for z/OS Administration Guide</i> .

Note: You can also use the Connect:Direct Browser User Interface to perform some of the procedures in this chapter. To learn more about the Connect:Direct Browser, see the documentation on the Connect:Direct Browser CD-ROM or available online from the Sterling Commerce Documentation Library.

The Primary Options Menu

The Primary Options Menu in the following figure contains the Connect:Direct functions which are available to most users.

```

node.name          Connect:Direct PRIMARY OPTIONS MENU
CMD ==>
Connect:Direct SIGNON process completed.
SELECT ONE OF THE FOLLOWING:
  CF - COPY A FILE *****
  SB - SUBMIT A PREDEFINED PROCESS *
  DF - DEFINE A PROCESS USING ISPF EDIT * TODAY: yyyy.mm.dd *
  SS - VIEW STATISTICS FOR A COMPLETED PROCESS * TIME: hh:mm *
  MB - SUBMIT A BATCH TO Connect:Enterprise for z/OS *
  CP - CHANGE CHARACTERISTICS OF A PROCESS *****
  DP - DELETE A NON-EXECUTING PROCESS
  FP - FLUSH AN EXECUTING PROCESS
  SP - VIEW DATA ABOUT AN EXECUTING PROCESS
  PS - SUSPEND AN EXECUTING PROCESS
  MSG - VIEW Connect:Direct MESSAGE TEXT
  SW - SWAP AMONG CONCURRENT Connect:Direct SESSIONS
  SD - VIEW/CHANGE YOUR Connect:Direct SIGNON INFORMATION DEFAULTS
  NM - VIEW INFORMATION IN THE Connect:Direct NETWORK MAP
  WHO - VIEW CHARACTERISTICS OF YOUR Connect:Direct IUI ENVIRONMENT
  SPF - ENTER ISPF/PDF
  AUTH - VIEW YOUR Connect:Direct FUNCTION AUTHORIZATION
  MS - SIGN ON TO MULTIPLE Connect:Direct NODES CONCURRENTLY
  ADMIN - PERFORM Connect:Direct ADMINISTRATIVE FUNCTIONS

```

To request a function, type its option identifier on the command line (CMD==>) and press **Enter**. The following options enable you to create and submit Processes to the TCQ:

Option	Description
CF	Displays the main Copy File Menu where you can set up a COPY Process.
SB	Displays the Submit Process screen where you can submit a predefined Process.
DF	Displays the Process Definition Screen where you can define or update a Process using the ISPF edit function and then submit the Process.

See Chapter 7, *Building, Modifying, and Submitting Processes*, for more information on these options.

The following option enables you to view statistics related to completed Processes:

Option	Description
SS	Displays the Select Statistics screen where you can examine the system statistics log records with various search criteria applied and select the output to go to a file, table, or printer. See Chapter 10, <i>Process Results and Statistics</i> , for more information on the Select Statistics screen.

The following options enable you to delete, flush, view, suspend, and change Processes in the TCQ:

Option	Description
CP	Accesses the Change Process screen where you can change the priority and class of a Process and the status of the Process in the TCQ. See Chapter 8, <i>Controlling Processes in the TCQ</i> , for more information on the Change Process screen.
DP	Displays the Delete Process screen where you can remove a nonexecuting Process from the TCQ.
FP	Displays the Flush Process screen where you can remove an executing Process from the TCQ.
SP	Displays the Select Process screen where you can view the status of Processes that you submitted (placed in the TCQ) for execution. See Chapter 8, <i>Controlling Processes in the TCQ</i> , for more information on the Select Process screen.
PS	Displays the Suspend Process screen where you can put an executing Process in the Hold TCQ. See Chapter 8, <i>Controlling Processes in the TCQ</i> , for more information on the Suspend Process screen.

The following options enable you to view your Connect:Direct environment:

Option	Description
MSG	Displays the Message Maintenance screen where you can display Connect:Direct messages or print a summary or full report of the Connect:Direct message file. See <i>Viewing and Printing Messages Using Different Criteria</i> on page 40, for more information on the Message Maintenance screen.
SD	Displays the Signon Defaults screen where you can examine and change your signon default values. See Chapter 6, <i>Managing Sessions</i> , for more information on the Signon Defaults screen.
NM	Displays the Select network map screen, where you choose for display or print the defined nodes from the network map file and translate TCP/IP host names and network addresses. See Chapter 11, <i>The Network Map</i> , for more information on the Select network map screen.

Option	Description
WHO	Displays the User Information screen, which indicates your user ID, user node, version, release, maintenance level of Connect:Direct and file information about your current session. See Chapter 6, <i>Managing Sessions</i> , for more information on viewing current sign on parameters.
AUTH	Displays the User Authorization screen. This screen displays commands you are authorized to use. See <i>Displaying Your User Authorization</i> on page 36 for more information.

The following options enable you to sign on and swap among multiple sessions:

Option	Description
MS	Displays the Multiple Session Signon screen where you can sign on to another DTF session. See page 67 for more information on the Multiple Session Signon screen.
SW	Displays the Swap/Display User Sessions screen where you can view all the multiple sessions that you are signed on to and swap sessions. See Chapter 6, <i>Managing Sessions</i> , for more information on the Multiple Session Signon screen.

The following options display other menus:

Option	Description
SPF	Displays an ISPF/PDF session, one level beneath the Connect:Direct panels. See <i>Navigating the Interactive User Interface</i> for more information on branching to an ISPF/PDF session.
ADMIN	Displays the Connect:Direct Administrative Options Menu. This menu displays a series of command options used for administering the system. See the <i>Connect:Direct for z/OS Administration Guide</i> for the menu produced by this option.

Navigating the Interactive User Interface

You can move from screen to screen with branch commands and function keys.

Using Branch Commands

Branch commands enable you to navigate quickly to a specific screen without returning to the Menu. You can type the branch command on the command line at the top of the screen (CMD==>) or on any input line preceded by ==>. To execute the branch, type = followed by the ID of the screen you want to branch to and press **Enter**.

For example, to go to the Select Process (SP) screen, type **=SP** on the command line and press **Enter**.

Note: The equal sign (=) is not required before an option on the command line of the Connect:Direct Signon screen or the Connect:Direct Primary Options Menu.

The following table describes two additional branch commands that you can use.

Command	Description
=ADMIN.xx where xx is the screen ID	Enables you to move to an administrative screen from a primary screen. For example, to branch from the WHO screen (which is an option of the Primary Menu) to the INQ screen (which is an option of the Administrative Menu), type =ADMIN.INQ at the command line of the WHO screen.
SPF.x where x is the SPF option that you want to display	The SPF option on the Primary Options Menu branches to the ISPF/PDF session, one level beneath the Connect:Direct panels. If you want to bypass the ISPF/PDF menu, type the SPF command with a period and the option. For example, to use the BROWSE function within ISPF/PDF, type SPF.1 on the command line. The SPF command is valid on each Connect:Direct screen command line.

Using the PF Keys

The PF keys listed in the following table are some of the programmed default definitions that you can use with Connect:Direct screens. The ISPF user-defined PF keys are supported.

Key	Result
PF1	Displays online field Help or long error message
PF3	Returns to the previous screen
PF4	Returns to the Primary Options Menu
PF7	Scrolls up
PF8	Scrolls down
PF10	Scrolls right
PF11	Scrolls left

Browsing the IUI Display

The TABLE or DISPLAY parameters which are options at the bottom of most Selection screens specify that the output of the selection command is stored in the Connect:Direct temporary file and then displayed upon successful completion of the command. The output is usually in tabular format. Use the commands, UP, RIGHT, or FIND, to browse the output. Press the **PF1** key to see extensive Help on how to browse the file.

Displaying Your User Authorization

Use the User Authorization screen to determine commands you are authorized to execute. To access the screen, select the **AUTH** option from the Primary Options Menu.

node.name	USER AUTHORIZATION				hh:mm
CMD ==>					
	AUTH	COMMAND	AUTH	COMMAND	
	-----		-----		
	1)	YES - CHANGE PROCESS	15)	YES - SELECT TASK	
	2)	YES - DELETE PROCESS	16)	YES - SELECT TYPE	
	3)	YES - DELETE TYPE	17)	NO - SELECT USER	
	4)	NO - DELETE USER	18)	YES - SUBMIT PROCESS	
	5)	YES - FLUSH PROCESS	19)	YES - SUBMIT WITHIN PROC	
	6)	YES - FLUSH TASK	20)	YES - SUSPEND PROCESS	
	7)	Y/Y - INSERT/UPDATE TYPE	21)	NO - STAT COMMAND	
	8)	N/N - INSERT/UPDATE USER	22)	NO - EVENT COMMAND	
	9)	YES - MODIFY (TRACE)	23)	YES - VIEW PROCESS	
	10)	YES - STOP Connect:Direct	24)	YES - PERFORM CRC OVERRIDES	
	11)	NO - UPDATE NETWORK MAP	25)	NO - CONFIRM DELETE	
	12)	NO - SELECT NETWORK MAP	26)	NO - CONFIRM DEL OFF	
	13)	YES - SELECT PROCESS	27)	NO - UPDATE APKEY	
	14)	YES - SELECT STATISTICS	28)	NO - SECURE+ ADMIN	
			29)	NO - UPDATE INITPARM	

YES or Y means you are authorized, NO or N means you are not authorized, and SUB means you are authorized only if you submitted the Process.

To branch to an authorized function, type the corresponding number on the command line.

Getting Help

Connect:Direct provides a Help facility which describes the parameters that you can type on the screen through the IUI. You can access the Help information in two ways:

- ◆ Type **HELP** on the command line and press **Enter**
- ◆ Press the **PF1** key

Press **Enter** to continue viewing the Help screens. When you have viewed the screens, you can either press **PF1** to continue to the tutorial or press **PF3** to return to the screen you branched from.

Operator Tables

The Operator Table is available only under the IUI. This display enables you to monitor and control Processes. It is accessed through the SELECT PROCESS command and provides a summary of PNODE Processes in the TCQ. See *Viewing and Controlling a Process through the Operator Table* on page 125 for a description and samples of the Operator Table.

```

node.name
Row 1 to 6 of 6

-----OPERATOR TABLE-----
==> Q
OPTION          PNAME          PNUMBER  SUBMITTER.NODE-- OTHER.NODE----- PAGE
SERVER          USERID
-----
          SEND100M      18        CD.NODE1      CD.STD45      EX  EX
          USER01
          SEND50K      20        CD.NODE2      10.20.129.151  ~P  EX
          USER01
          COPYPAY      25        CD.NODE3      CD.STD44      ~P  EX
          USER01
          BKUPDSK1      2         CD.NODE4      10.20.129.141  ~P  EX
          USER01
          BENCH50      13,218    CD.NODE5      CD.STD45      ~P  EX
          USER01
          COPYCF      21        CD.NODE6      CD.STD45      HO  HI
          USER01
***** Bottom of data *****

```

You can press **Enter** to refresh the display and monitor the progress of the Processes displayed on the screen. The following table describes the commands that you can issue from the Operator Table.

Command	Description
H	Hold
D	Delete
R	Release
P	Suspend
F	Flush
V	View Processes
S	Show Detail

Type the option that you want in the **OPTION** column next to the name of the Process as shown in the previous screen.

Note: The Executing Queue version of the Operator Table displays the executing Processes only; therefore, the options available to it are limited to the Suspend (P) and Show detail (S) only.

Messages

You may encounter two types of messages: ISPF/PDF messages and Connect:Direct messages.

Message Type	Description
ISPF/PDF messages	Display at the top of the screen when the value you typed for a field does not pass the editor. You get this kind of message if, for example, you type an alphabetic character in a numeric field. Press PF1 to see a longer explanation of the error or type HELP on the command line and press Enter .
Connect:Direct messages and sense codes	Contain a short and long text which explain an error you get when you submit your command. This type of error is indicated by a message ID with the format xxxnnnx, for example, SVSL003I. You get this error, if, for example, you type an invalid Process name on a SELECT PROCESS command. Connect:Direct generates a sense code to indicate a connection error involving a netmap. The format for a sense code is SENSnnnn where nnnn is the four-digit number representing the sense code.

Message Example

In the following example, the user typed ABC for Process Number, a numeric field, resulting in the message PROCESS NUMBER in the upper right corner. Pressing **PF1** produces the longer message *"ABC" IS INVALID...*

```

node.name                SELECT PROCESS                PROCESS NUMBER
CMD ==> 0                hh:mm
"ABC" IS INVALID; MUST BE IN THE RANGE 0 - 199,999.
  CMD:  O ... OPERATOR TABLE  S ... OPERATOR TABLE/EXEC QUEUE STATUS
        P ... PRINT REPORT      D ... DISPLAY REPORT      V ... VIEW PROCESS

QUEUE:                    ==> _ (A-ALL,W-WAIT,E-EXECUTE,H-HOLD,T-TIMER)

PROCESS NUMBERS: ==> _____ ==> _____ ==> _____ ==> _____
PROCESS NAMES:   ==> _____ ==> _____ ==> _____ ==> _____
SERVER NAMES:    ==> _____ ==> _____ ==> _____ ==> _____

STATUS:          (HO,HR,HI,HE,HC,HP,HS,RH,RA,WC,WX,WT,H,R,W)
                ==> _      ==> _      ==> _      ==> _

DESTINATION NODES:
                ==> _____ ==> _____
                ==> _____ ==> _____

USER ID:         NODE ID:
==> _____
                ==> _____
==> _____
                ==> _____

DO YOU WANT VALUES FOR THIS REQUEST TO BE CASE SENSITIVE? ==> NO

```

The following figure is an example of the short text message that is displayed at the top of the screen.

```

No process(es) found matching the search criteria.

```

The long text provides a more detailed explanation of the message and can also include the system action and a suggested response. Press **PF1** to see the long explanation of the error or type **HELP** or **M** on the command line and press **Enter**. The following figure shows a sample message.

```

node.name                Connect:Direct MESSAGE DISPLAY                hh:mm
CMD ==>

MSGID ==> SVSL003I
MODULE ==> DMVSOPEN

Copy requested DISP=(,CATLG) to already cataloged dataset.

The PROCESS COPY step requested a DISP=(,CATLG) on the TO
clause of the COPY statement. The requested dataset already
exists as a cataloged dataset.

System Action. The PROCESS COPY step is terminated with a
completion code of 8.

Response: Either correct the COPY dataset or uncatalog the
existing dataset and re-submit the PROCESS.

```

The Statistics Summary screen, an option of the SELECT STATISTICS command, also gives you the option of branching to the message screen associated with a failed Process. Type an **M** next to the Process name which failed (indicated by an *), and Connect:Direct displays the extended message.

Messages in a Connect:Direct/Plex Environment

In a Connect:Direct/Plex environment, messages can originate from any Connect:Direct Server or from the Connect:Direct Manager. The system administrator can define a 2-character message ID that identifies which Connect:Direct/Plex member originated the message. This message ID is displayed after the message number, as in the following example. The message ID is highlighted in bold in this example.

SVTM055I	s1	SESSION (001) ESTABLISHED WITH	SNODE=SC.DUB.OS390
SVTM055I	s1	SESSION (001) ESTABLISHED WITH	PNODE=SC.DUB.OS390
SVTM036I	s1	PROCESS STARTED MVS2MVST(1) PNODE=SC.DUB.OS390
SVTM036I	s1	PROCESS STARTED MVSM2VST(1) SNODE=SC.DUB.OS390

The system administrator defines the message ID using the CDPLEX.MSGID initialization parameter, as described in the *Customizing Connect:Direct* chapter of the *Connect:Direct for z/OS Administration Guide*.

Viewing and Printing Messages Using Different Criteria

You can use the Message Maintenance screen to view or print Connect:Direct Message Text. You can display messages by message ID if you need to look up an error message in your Process statistics, or if you have a batch job that failed.

1. From the Primary Options Menu, select option **MSG**.

Note: The MSG command is available only through the IUI.

The following sample screen is a combination of two panels to show the prefixes used in Connect:Direct messages.


```

Connect:Direct MESSAGE LOOKUP          2 Members processed
                                         DATE => yyyy/mm/mm
                                         TIME => hh:mm
ENTER OPTION ==> _
=====
                                OPTIONS
                                =====
1 = DISPLAY MESSAGE TEXT BY MESSAGE ID
2 = DISPLAY LIST OF ENTRIES (OPTIONALLY BY MODULE)
  MODULE==> _____
3 = PRINT SUMMARY REPORT
4 = PRINT FULL REPORT

The following is the general basis for C:D messages:

      MAINFRAME: z/OS, OS/390, VM, VSE, Security (ACF2, RACF, TSS), etc.
MESSAGE          MESSAGE
PREFIX    Applies to:      PREFIX    Applies to:

N xxx      CICS API failures      Q xxx      CICS API failures
RACF xxxxx Security system MSGs   S xxxxxxxx C:D Mainframe msgs
SENS nnnn  NETMAP sense code     U nnnn    z/OS User ABEND code
CSPx xxxx  Secure Plus           MBSx xxxx Multi-Byte Character Set

                                NON-Mainframe
MESSAGE          MESSAGE
PREFIX    Applies to:      PREFIX    Applies to:

A xxxxxxxx i5/OS or OS/400       CCUP xxxxx Control Center
EMSL xxxxx HP NonStop or Tandem  FSLD xxxxx HP NonStop or Tandem
IOXT xxxxx HP NonStop or Tandem  J xxxxxxxx C:D for the Web
L xxxxxxxx Windows, SI, or Select V xxxxxxxx OpenVMS
X xxxxxxxx Linux, UNIX, or Stratus

```

2. Type **1** on the command line and press **Enter**. The Message Display Screen appears.

Type the message ID and press **Enter**. (To check a sense code, type **SENSnnnn** where *nnnn* is the four-digit number representing the sense code and press **Enter**.) The full text of the message (or sense code) is displayed. In the following example, the information associated with Message SCBI190I is shown.

```

Connect:Direct MESSAGE DISPLAY
                                         DATE => yyyy.mm.dd
                                         TIME => hh:mm

MESSAGE ID==> SCBI190I
MODULE      ==> DMCBSUBM
=====

SHORT TEXT==> Process specified not in process library.
LONG TEXT:
  LINE 1 ==> This message may be generated for the following reasons:
  LINE 2 ==>
  LINE 3 ==> 1) The member specified is not in the public process library
  LINE 4 ==>    or in the PDS specified in the 'DSN=' keyword.
  LINE 5 ==> 2) The process library has not been properly concatenated.
  LINE 6 ==>
  LINE 7 ==>
  LINE 8 ==>
  LINE 9 ==>
  LINE 10==> SYSTEM ACTION: Return to invoker with RC=8.
  LINE 11==>
  LINE 12==> RESPONSE: Verify the above is correct and resubmit process.

COMMAND  ==> _____ ENTER 'DIR' TO DISPLAY THE DIRECTORY

```

3. Type **DIR** at the COMMAND prompt at the bottom of the screen to display the directory of the module messages.

Viewing a Message List

You can display a list of all Connect:Direct messages or a list of messages generated by a module.

1. From the Primary Options Menu, select option **MSG**.

Note: The MSG command is available only through the IUI.

2. Type **2** on the command line, specify a Module (optional), and press **Enter**. The Message Summary screen appears.

Note: If you do not specify a module name, all messages are displayed.

```

-----Connect:Direct MESSAGE SUMMARY----- Row 1 of 13169
CMD ==>                                     SCROLL ==> PAGE
MSG ID  MODULE  MESSAGE
AAPR100I ap_check Call to AP_PARSE_OUT failed.
AAPR101I dtf_ap_c C:D-400 License will expire in &DAYS days.
AAPR500I ap_parse Error parsing keyword.
AAPR501I ap_parse Error in parser while parsing NDM$$KEY value.
AAPR502I ap_parse End of NDM$$KEY value encountered to soon.
AAPR503I ap_parse AP keyword too long.
AAPR504I ap_parse Error parsing CPU value.
AAPR505I ap_parse Error parsing PN value.
AAPR506I ap_parse Error parsing ESF value.
AAPR507I ap_parse SDF value not in list form.
AAPR508I ap_parse SDF list entry invalid.
AAPR509I ap_parse Parse error during SDF list processing.
AAPR510I ap_parse Error parsing SNA value.
AAPR511I ap_parse Error parsing EIO value.
AAPR512I ap_parse Error parsing PID value.
AAPR514I ap_parse Error parsing SEQ value.
AAPR516I ap_parse Error parsing expiration date value.
AAPR518I ap_parse Error parsing NETEX value.
AAPR519I ap_parse Unknown keyword.
AAPR521I ap_parse No value specified for PID.
AAPR522I ap_parse No value specified for SEQ.

```

3. Type an **S** next to the message ID for the full Message Display screen.

Printing Messages

You can print the Connect:Direct messages file in summary form or in detail, including the long form of the message text.

1. From the Primary Options Menu, select option **MSG**.

Note: The MSG command is available only through the IUI.

2. Do one of the following:
 - ◆ Type **3** on the command line for a summary report, and press **Enter**.
 - ◆ Type **4** on the command line for a detail report, and press **Enter**.

Specify the sysout class for the message file print output and the ID for a remote printer. If you want your output to go to a preallocated data set instead, specify the name of the data set. You must enclose the data set name in single quotes and give it the following DCB attributes.

```
DSORG=PS, RECFM=FB, LRECL=80, BLKSIZE=0
```

The Application Program Interface

This chapter describes how to create an interface with Connect:Direct for z/OS. The following topics are covered:

- ◆ Communicating with the Connect:Direct API
- ◆ The DMCHLAPI program
- ◆ Examples of DMCHLAPI parameters and calling sequences

Communicating with the Connect:Direct API

The high-level application-program, DMCHLAPI, handles communication between the Connect:Direct API and any application program that provides Connect:Direct with command strings for batch processing. You can write an application program following the rules described in the following sections.

You can also process Connect:Direct command strings by using the DMBATCH batch interface program. For more information on using DMBATCH, refer to Chapter 3, *The Batch Interface*.

Both DMCHLAPI and DMBATCH are in \$CD.LINKLIB.

You can design user-written applications that have an interface to Connect:Direct. You can write the applications in any computer language, including PL/I, Assembler, and COBOL. The following samples are in the \$CD.SAMPLIB library:

Application Name	Description
ASMSAMP	Assembler language user-written application, called
PLISAMP	PL/I language user-written program, called
COBSAMP	COBOL language user-written program, called

Note: All API programs run in 31-bit mode and they must call DMCHLAPI in 31-bit mode. You can either link your program as AMODE 31, or establish 31-bit addressability before calling DMCHLAPI. See the sample program ASMSAMP for an example of establishing 31-bit addressability and returning to 24-bit addressability after the call to DMCHLAPI. Also execute DMCHLAPI with a BASSM rather than a BALR. Macro GENCALL is provided in the sample library to perform a BASSM. See the sample program ASMSAMP for an example.

The DMCHLAPI Program

The DMCHLAPI program communicates with the API through a control block interface called the User Interface Control Block (UICB). DMCHLAPI works in the following sequence:

1. DMCHLAPI accepts Connect:Direct command strings from an application program and passes the strings to the API.
2. The user-written application requests DMCHLAPI to perform output formatting routines after returning from the API. These formatting routines display information about the Connect:Direct command that just completed processing.
3. After execution of each command, DMCHLAPI issues a return code reflecting the status of API communications.

Required Parameters

DMCHLAPI requires Connect:Direct to pass three parameters on every invocation from an application program. The parameters, in order, are:

Parameter	Description
CMDLEN (first parameter)	Points to a variable-length character string (up to 4096 bytes) that contains the string length in the first halfword and the command text in the remainder of the string. Just as in DMBATCH, the first command string must be a SIGNON and the last command must be a SIGNOFF. The following figure shows the command string format.

Length of Command String	Command String
H'32'	SELECT PROC WHERE(PNUM=02)
	132.... 32

Parameter	Description																																															
UICB@ (second parameter)	Points to a fullword pointer of zeroes, which is filled in with the UICB address by DMCHLAPI when DMCHLAPI receives the SIGNON command from the application program. Information in the UICB can now be interrogated by the application program, if necessary. The UICB@ field is cleared and filled again with zeroes when DMCHLAPI receives the SIGNOFF command.																																															
OUTSPECS (third parameter)	<p>Points to a 7-character string containing the output format specifications. The output is written to the file name defined by DMPRINT. Each specification is one character long. Refer to the following descriptions.</p> <table border="1"> <thead> <tr> <th>Field</th> <th>Values</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td rowspan="2">1</td> <td>Y</td> <td>Displays the command string that executed.</td> </tr> <tr> <td>N</td> <td>Does not display the command string that executed.</td> </tr> <tr> <td rowspan="2">2</td> <td>Y</td> <td>Displays the API return code and message ID.</td> </tr> <tr> <td>N</td> <td>Does not display the API return code and message ID.</td> </tr> <tr> <td rowspan="3">3</td> <td>S</td> <td>Displays the short message text when a return code of zero comes from the API.</td> </tr> <tr> <td>L</td> <td>Displays long message text when a return code of zero is received from the API.</td> </tr> <tr> <td>N</td> <td>Does not display message text when a return code of zero is received from the API.</td> </tr> <tr> <td rowspan="3">4</td> <td>S</td> <td>Displays the short message text when a nonzero return code comes from the API.</td> </tr> <tr> <td>L</td> <td>Displays long message text when a nonzero return code comes from the API.</td> </tr> <tr> <td>N</td> <td>Does not display message text when a return code greater than zero comes from the API.</td> </tr> <tr> <td rowspan="2">5</td> <td>Y</td> <td>Displays the data generated in the temporary file.</td> </tr> <tr> <td>N</td> <td>Does not display the data generated in the temporary file.</td> </tr> <tr> <td rowspan="2">6</td> <td>Y</td> <td>Displays the string that identifies UICB fields which the Extract feature returns information about.</td> </tr> <tr> <td>N</td> <td>Does not display the string that identifies UICB fields which the Extract feature returns information about.</td> </tr> <tr> <td rowspan="2">7</td> <td>Y</td> <td>Displays a dump of the area that received output from the Extract feature.</td> </tr> <tr> <td>N</td> <td>Does not display a dump of the area that received output from the Extract feature.</td> </tr> <tr> <td rowspan="2">8</td> <td>Y</td> <td>Prints detailed SELECT STATISTICS if MAXDELAY specified and Process fails.</td> </tr> <tr> <td>N</td> <td>Does not print SELECT STATISTICS if MAXDELAY specified and Process fails.</td> </tr> </tbody> </table> <p>All output generated as a result of these specifications is routed to a DDNAME of DMPRINT. No output is created if the DDNAME is not present.</p> <p>The following example shows the most common specifications for this parameter.</p> <pre>OUTSPECS DC C'YYSLYNNN'</pre> <p>Positions 6 and 7 are not used by DMCHLAPI unless the extract feature of DMCHLAPI is used, as explained in the next section.</p>	Field	Values	Meaning	1	Y	Displays the command string that executed.	N	Does not display the command string that executed.	2	Y	Displays the API return code and message ID.	N	Does not display the API return code and message ID.	3	S	Displays the short message text when a return code of zero comes from the API.	L	Displays long message text when a return code of zero is received from the API.	N	Does not display message text when a return code of zero is received from the API.	4	S	Displays the short message text when a nonzero return code comes from the API.	L	Displays long message text when a nonzero return code comes from the API.	N	Does not display message text when a return code greater than zero comes from the API.	5	Y	Displays the data generated in the temporary file.	N	Does not display the data generated in the temporary file.	6	Y	Displays the string that identifies UICB fields which the Extract feature returns information about.	N	Does not display the string that identifies UICB fields which the Extract feature returns information about.	7	Y	Displays a dump of the area that received output from the Extract feature.	N	Does not display a dump of the area that received output from the Extract feature.	8	Y	Prints detailed SELECT STATISTICS if MAXDELAY specified and Process fails.	N	Does not print SELECT STATISTICS if MAXDELAY specified and Process fails.
Field	Values	Meaning																																														
1	Y	Displays the command string that executed.																																														
	N	Does not display the command string that executed.																																														
2	Y	Displays the API return code and message ID.																																														
	N	Does not display the API return code and message ID.																																														
3	S	Displays the short message text when a return code of zero comes from the API.																																														
	L	Displays long message text when a return code of zero is received from the API.																																														
	N	Does not display message text when a return code of zero is received from the API.																																														
4	S	Displays the short message text when a nonzero return code comes from the API.																																														
	L	Displays long message text when a nonzero return code comes from the API.																																														
	N	Does not display message text when a return code greater than zero comes from the API.																																														
5	Y	Displays the data generated in the temporary file.																																														
	N	Does not display the data generated in the temporary file.																																														
6	Y	Displays the string that identifies UICB fields which the Extract feature returns information about.																																														
	N	Does not display the string that identifies UICB fields which the Extract feature returns information about.																																														
7	Y	Displays a dump of the area that received output from the Extract feature.																																														
	N	Does not display a dump of the area that received output from the Extract feature.																																														
8	Y	Prints detailed SELECT STATISTICS if MAXDELAY specified and Process fails.																																														
	N	Does not print SELECT STATISTICS if MAXDELAY specified and Process fails.																																														

Extracting Return Codes

If you want the Return Code from the processed Connect:Direct command, you must use the extract feature. This feature enables you to extract certain UICB fields after Connect:Direct command execution.

To activate the optional extract feature, the program must pass the following additional parameters to DMCHLAPI:

Parameter	Description															
EXTSTRLN (fourth parameter)	<p>The UICB extract string. The user-supplied application must set up an extract string identifying the UICB fields to extract. The string consists of a halfword length field containing the length of the extract string (excluding the itself halfword length field), followed by the first UICB field to extract, a space, the second UICB field to extract, a space, and so on.</p> <p>A sample extract string follows.</p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="border: 1px solid black; width: 10%; padding: 2px;">H'32'</td> <td style="border: 1px solid black; padding: 2px;">UITMPDDN UIRTNCD UIPROC# UIMSGID</td> </tr> </table> </div> <p>You can only specify the UICB fields listed in the extract string. If you encounter an invalid keyword in the string, the extract routine terminates execution, and no more information is extracted. The application is informed of the error by means of the EXTRC, EXTMSG, and INVALIDKEY parameters. If field 6 of the OUTSPECS parameter is set to Y, the extract string is written to the file defined by DMPRINT.</p>	H'32'	UITMPDDN UIRTNCD UIPROC# UIMSGID													
H'32'	UITMPDDN UIRTNCD UIPROC# UIMSGID															
EXTAREA (fifth parameter)	<p>Where the UICB extracted information is placed. The application is responsible for ensuring that the area is large enough to accommodate the information requested by the extract string. If field 7 of OUTSPECS is set to Y, the extract string is written to the file defined by DMPRINT. The following shows how to calculate required storage for the extract area necessary for the extract string in the previous figure.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>UICB Field</th> <th>Data Definitions</th> <th>Storage</th> </tr> </thead> <tbody> <tr> <td>UITMPDDN</td> <td>Character length of 8</td> <td>8 bytes</td> </tr> <tr> <td>UIRTNCD</td> <td>Fullword</td> <td>4 bytes</td> </tr> <tr> <td>UIPROC#</td> <td>Fullword</td> <td>4 bytes</td> </tr> <tr> <td>UIMSGID</td> <td>Character length of 8</td> <td>8 bytes</td> </tr> </tbody> </table>	UICB Field	Data Definitions	Storage	UITMPDDN	Character length of 8	8 bytes	UIRTNCD	Fullword	4 bytes	UIPROC#	Fullword	4 bytes	UIMSGID	Character length of 8	8 bytes
UICB Field	Data Definitions	Storage														
UITMPDDN	Character length of 8	8 bytes														
UIRTNCD	Fullword	4 bytes														
UIPROC#	Fullword	4 bytes														
UIMSGID	Character length of 8	8 bytes														
EXTRC (sixth parameter)	A 4-byte binary field containing the extract feature return code.															
EXTMSG (seventh parameter)	An 8-character field containing an 8-character message ID from the extract feature.															
INVALIDKEY (eighth parameter)	An 8-character field containing an invalid extract string keyword. DMCHLAPI relays the location of the extract string error to the application in this parameter. This field contains the invalid item in the extract string if EXTRC is nonzero.															

DMCHLAPI specifies either the first three or all eight parameters. If an incorrect number of parameters is passed, DMCHLAPI issues an error message and assigns a return code of 20. Processing cannot occur during this time.

UICB Fields

The following table shows the valid UICB fields. Specifying these fields in the EXTSTRLN parameter results in feedback in the extract area, defined by the EXTAREA parameter. Exact definitions are in the DMUICB macro found in SAMPLIB. In the table, the Control Block Builder Syntax Error Work Areas have a value only when appropriate. For the Boolean Flags, output is Y (bit is on) or N (bit is off).

Name	Type	LN	Description
UIRCBLNG	halfword	2	The length of the control block
UIDESCR	character	16	UICB identifier
UITCA	address constant	4	Task Control Area
UIBRCB	address constant	4	Batch Region Control Block
UIDYNCB	address constant	4	Dynamic Allocation Control Block
UITPCB	address constant	4	Text Parser Control Block
UILEVEL	fullword	4	Modal level counter
UITMPDCB	address constant	4	Temporary file DCB
UIMSGCB	address constant	4	Message Control Block
UIUNODE	character	16	User node ID
UIUID	character	64	User ID
UIPSWD	character	64	Signon password
UINPSWD	character	64	New Signon password
UITMPVOL	character	6	Volume serial number of temporary file if not specified by user
UNTMPDDN	character	8	DDNAME used for temporary DSN
UIUSRTYP	character	1	User Type (operator, administrator, user)
UIAPPLID	character	8	VTAM logon ID
UIRTNCD	fullword	4	API return code
UIMSGID	character	8	API message ID
UILNODE	character	16	Name of node that is "local" to DTF
UILPP	halfword	2	Lines per page for printed output
UITMPLNG	halfword	2	Length of TSO temporary file name
UITMPDSN	character	44	TSO temporary file name
UIPUBLNG	halfword	2	Length of Process Library name
UIPUBDSN	character	44	Process Library name

Name	Type	LN	Description
UIMSGLNG	halfword	2	Length of message library name
UIMSGDSN	character	44	Message library name
UINETMAP	character	64	Network map file name
UIPROC#	fullword	4	Process number from submit
UIDSPY	address constant	4	Address of SCDSPLY
UISTRING	address constant	4	Address of SCSTRING
*****	*****		Start of control block builder syntax error work area
UILABL#	halfword	2	Length of Connect:Direct label
UILABL	character	8	Connect:Direct label
UICMD1#	halfword	2	Length of first word in Connect:Direct command
UICMD1	character	8	First word in Connect:Direct command
UICMD2#	halfword	2	Length of second word in Connect:Direct command
UICMD2	character	8	Second word in Connect:Direct command
UIKLST#	halfword	2	Length of keyword that starts a list
UIKLST	character	8	Keyword that starts a list
UIKEYW#	fullword	4	Length of keyword in list before error
UIKEYW	character	8	Last keyword in list before an error
UIPARM#	halfword	2	Length of parameter associated with UIKEYW
UIPARM	character	8	Parameter in error associated with UIKEYW
UIERRM1#	halfword	2	Length of UIERRM1 string
UIERRM1	character	64	All of the above work areas resolved into a string
UIERRM2	character	64	Msg ID and text for parsing error
*****	*****		Start of boolean flags
UIERRON	character	1	Indicates message in UIERRM1
UIERRLAB	character	1	Indicates something in UILABL
UIERRCM1	character	1	Indicates something in UICMD1
UIERRCM2	character	1	Indicates something in UICMD2
UIERRSCP	character	1	Indicates parsing error
UIERRC1O	character	1	Indicates open delimiter after command keyword, for example, "IF ("
UIERRC2O	character	1	Indicates open delimiter after second command keyword

Name	Type	LN	Description
UIERRG	character	1	Indicates VTAM error msg in UIERRM1
UIERRLST	character	1	Indicates something in UIKLST
UIERRLOP	character	1	Indicates open delimiter after UIKLST
UIERRLCL	character	1	Indicates close delimiter after UIKLST
UIERRLEQ	character	1	Indicates equal sign after UIKLST
UIERRLCM	character	1	Indicates comma after UIKLST
UIERRLSP	character	1	Indicates space after UIKLST
UIERRKEY	character	1	Indicates something in UIKEYW
UIERRKOP	character	1	Indicates open delimiter after UIKEYW
UIERRKCL	character	1	Indicates close delimiter after UIKEYW
UIERRKEQ	character	1	Indicates equal sign after UIKEYW
UIERRKCM	character	1	Indicates comma after UIKEYW
UIERRKSP	character	1	Indicates space after UIKEYW
UIERRBPC	character	1	Indicates close delimiter before a parameter in a list
UIERRPRM	character	1	Indicates something in UIPARM
UIERRPOP	character	1	Indicates open delimiter after UIPARM
UIERRPCL	character	1	Indicates close delimiter after UIPARM
UIERRPEQ	character	1	Indicates equal sign after UIPARM
UIERRPCM	character	1	Indicates comma after UIPARM
UIERRPSP	character	1	Indicates space after UIPARM
UIF1SUBM	character	1	Indicates a submitted Process
UIGOTDSN	character	1	Indicates found temporary file name as Signon command parameter
UITFILE	character	1	Indicates data is generated into temporary file
UIEOF	character	1	Indicates reached EOF of Process file
UIMODAL	character	1	Indicates modal statement processed
UITFILEX	character	1	Reserved
UIMASTER	character	1	Indicates this UICB is master
UIRECON	character	1	Indicates a reconnect attempted
UIINACT	character	1	Indicates VTAM session for this UICB failed
UILOCAL	character	1	Indicates local node session

Name	Type	LN	Description
UIESF	character	1	Indicates ESF=YES on Signon command
UIZOPSWD	character	1	Indicates blank password on Signon
UIZNPSWD	character	1	Indicates blank new password on Signon
*****	*****		END OF BOOLEAN FLAGS
UIRAT@	address constant	4	PTR to resource address table
UIDRLSE#	halfword	2	DTF release level
UIDPUF#	halfword	2	DTF PUF level
UIDPUT#	halfword	2	PUT tape number
UILINE#	halfword	2	Line number within Process in error
UIRLSE#	halfword	2	Current release, version and mod level
UIPUF#	halfword	2	Current PUF level
UI@MASTR	address constant	4	Master (user) UICB
UI@ACTIV	address constant	4	Currently active UICB
UI@FPTR	address constant	4	Next UICB
UI@BPTR	address constant	4	Previous UICB
UIALOTYP	character	8	Allocation type for temporary file
UIALOPRI	character	8	Allocation of prime space for temporary file
UIALOSEC	character	8	Allocation of secondary space for temporary file
UIALOUNI	character	8	Allocation unit for temporary file
UIALOVOL	character	8	Allocation of volume serial number for temporary file
UIFOLD	character	3	Fold to upper case if YES UIPRTALC
UIPRTALC	character	80	User-defined Connect:Direct print destination
UIPACCT#	halfword	2	Length of PNODE accounting data
UIPACCT	character	255	PNODE accounting data text
UISACCT#	halfword	2	Length of SNODE accounting data
UISACCT	character	255	SNODE accounting data text
UIGSCAPI (VM)	character	8	GCS API virtual machine identifier

DMCHLAPI Return Codes

Return codes reflect the status of the DMCHLAPI communications with the API. These codes only reflect whether DMCHLAPI could process the request and pass the command to the DTF. They do not reflect the completion status of the command.

The following table defines each return code.

RC	Description
00000000	The command executed normally.
00000004	Signon to the master session failed, but the ESF environment is established.
00000008	A non-ESF command attempted in an ESF environment.
0000000C	A session is lost in a multiple session environment.
00000010	The master session is lost.
00000014	The master session signed off successfully.
00000018	The master session signon failed, and no ESF ability exists.
0000001C	A non-master signon failed.
00000020	DMCHLAPI received an invalid number of input parameters.
00000024	The output specifications included an invalid parameter.
00000028	An invalid pointer to the UICB is passed to DMCHLAPI.

For more information on master sessions and multiple session environments, refer to Chapter 6, *Managing Sessions*.

Extract Fields for DMCHLAPI Return Codes

The following extract fields apply to DMCHLAPI return codes:

Field	Description
UIRTNCD	Lists return codes. UIRTNCD is set on completion of every command processed by the DTF.
UIMSGID	Lists message IDs. UIMSGID is set on completion of every command processed by the DTF.
UIPROC#	Identifies Process numbers. UIPROC# is set for every successful Submit command.

See *UICB Fields* on page 49 for the complete list of UICB fields.

DMCHLAPI Examples

The examples in this section show how required and optional parameters appear in sample programs. There is also a sample job stream for executing the program.

Required Parameters and Calling Sequences

The following example shows how the required parameters and calling sequences appear in a sample Assembler program.

```

*****
*   THE DATA DEFINITIONS ARE AS FOLLOWS:   *
*****

CMDLEN    DS    H           LENGTH OF COMMAND STRING
CMD       DS    CL4096     COMMAND STRING
UICB@     DS    A           POINTER TO UICB
OUTSPECS  DC    C'YYSLYNN' OUTPUT SPECIFICATIONS

*****
*
*   THE DMCHLAPI INVOCATION WOULD BE:       *
*
*****
CALL DMCHLAPI, (CMDLEN, UICB@, OUTSPECS), VL

```

The following example shows how the required parameters and calling sequences appear in a sample PL/I program.

```

*****
*   THE DATA DECLARATIONS ARE AS FOLLOWS: *
*****
DECLARE
CMD_PARM      FIXED BINARY (31) BASED (R),
UICB_PARM     FIXED BINARY (31) INITI (0'B'),
OUTSPEC_PARM  CHAR (7) INIT ('YYSLYNN'),
CMD           CHAR (4096) VARYING,
R             POINTER;

*****
*
*   THE DMCHLAPI INVOCATION WOULD BE:       *
*
*****
R=ADDR(CMD) ;
CALL DMCHLAPI (CMD_PARM, UICB_PARM, OUTSPEC_PARM);

```

Optional Parameters

The following example shows how the optional parameters look in a sample Assembler program.

```

*****
**      THE PARAMETERS WOULD BE DEFINED AS FOLLOWS:      **
*****

CMDLEN   DS   H
CMD      DS   CL4096
UICB@    DS   A
OUTSPECS DC   C'YYSLYYY'
EXTSTRLN DC   H'32'
EXTSTR   DC   C'UITMPDDN UIESF UIPROC# UIMSGID'
EXTAREA  DS   OCL24 /*OUTPUT FROM THE UICB EXTRACT          */
EXTMPDDN DS   CL8 /*VALUE OF UITMPDDN                      */
EXESF    DS   F /*VALUE OF UIRTNCD                          */
EXPROC#  DS   XL4 /*VALUE OF UIPROC#                                       */
EXMSGID  DS   CL8 /*VALUE OF UIMSGID                                       */
EXTRC    DS   F
EXTMSG   DS   CL8
INVALKEY DS   CL8

*****
**      NOTE THAT THE EXTRACT STRING ITSELF IS 30 BYTES LONG, *
**      BUT THE DUMP OF ITS EXTRACT WILL USE 31 BYTES.      *
**                                                         *
**      INSIDE THE MAIN BODY OF THE PROGRAM, THE CALL TO   *
**      DMCHLAPI WOULD LOOK LIKE:                          *
**                                                         *
*****

CALL DMCHLAPI, (CMDLEN, UICB@, OUTSPECS, EXTSTRLN, EXTAREA,
               EXTRC, EXTMSG, INVALKEY), VL

```

Sample Job Stream for Executing the Program

The following example shows a sample job stream that executes a Connect:Direct program that invokes DMCHLAPI.

```

//jobname JOB (ACCT), 'PGMR NAME',
//      NOTIFY=TSOUER, TIME=(1),
//      MSGCLASS=X, CLASS=A, PRTY=9, REGION=1024K
//JOBLIB   DD   DISP=SHR, DSN=$CD.LINKLIB
//CD       EXEC PGM=USERPROG
//DMPRINT  DD   SYSOUT=*
//DMNETMAP DD   DISP=SHR, DSN=$CD.VSAM.NETMAP
//DMPUBLIB DD   DISP=SHR, DSN=$CD.PROCESS
//DMMSGFIL DD   DISP=SHR, DSN=$CD.VSAM.MESSAGE
//SYSUDUMP DD   SYSOUT=*
//NDMCMD5 DD   SYSOUT=*

```

You must allocate several files using DD statements before executing a Connect:Direct application program that calls DMCHLAPI.

The following table defines the DD statements in the sample job stream:

Statement	Definition
DMPRINT	All output from SELECT PROCESS, SELECT STATISTICS, SELECT TYPE, and SELECT USER commands is written to this DD. Also, output as a result of the OUTSPECS specification is written to DMPRINT. DMPRINT is optional, but often required.
DMNETMAP	This definition is the name of the network map file that contains the names of all the nodes that Connect:Direct communicates with. DMNETMAP is required if no NETMAP keyword is specified on the SIGNON command.
DMPUBLIB	This definition is a library containing Processes. It is required.
DMMSGFIL	This definition is the name of the Message file. It is required.
NDMCMDS	If allocated, all command strings are written to the current file. It is optional.

Managing Sessions

This chapter provides the information you need to perform the following tasks:

- ◆ Signing on to Connect:Direct for z/OS
- ◆ Signing on to multiple sessions
- ◆ Swapping nodes
- ◆ Signing off of Connect:Direct
- ◆ Sequencing the SIGNON, SWAP NODE, and SIGNOFF commands

Each section describes the command format, parameters, batch interface, and Interactive User Interface. Examples are given in the batch sections.

Signing On to Connect:Direct for z/OS

The SIGNON command establishes your session with the DTF so that you can issue other commands.

You can control access to the Connect:Direct system through the Connect:Direct Authorization Facility and through security exits. Refer to *Implementing Security in Connect:Direct for z/OS Administration Guide* for information on security control.

If you are using the Extended Submit Facility (ESF), you can submit the SIGNON command even though the DTF is inactive.

Timeout Problems

If you are logged off involuntarily while using Connect:Direct via TCP/IP, your problem may be related to the TCP/IP connection not being able to communicate with the DTF. If you see the message *TCP/IP SEND error; Connection Lost*, and press **F1** to see the help for this message, the following text is displayed for message SVTC006I:

```
MSGID ==> SVTC006I
MODULE ==> DMTSTSND
```

```
TCP/IP SEND error; Connection Lost
```

```
An error has been detected while sending a command via TCP/IP.
The TCP/IP connection has been lost. A possible cause is
Connect:Direct has been shutdown and all TCP/IP connections
have been closed.
```

```
System Action: Normal processing can not continue.
```

```
Response: Logoff and attempt to signon again.
```

Although the TCP/IP protocol does not have a mechanism to send you a console message indicating that the client has been dropped and your session has timed out, you can determine if this has happened to you by looking in the statistics file for a signoff record with an SAFA019I error message that matches your user ID (be sure to enter **Y** in the CHANGE EXTENDED OPTS field on the SELECT STATISTICS panel and then enter **SO** as a record type on the next screen). For more information, see *Signon and IUI/API Errors* in the *Connect:Direct for z/OS Administration Guide*.

Connect:Direct provides an inactivity timer that is based on the number of minutes a user can remain inactive without communicating with the DTF. An administrator can specify this value using the TCP.API.TIMER global initialization parameter. Connect:Direct will only recognize changes made to the TIMEOUT parameter by an administrator. If you attempt to change this parameter and you are not an administrator, you will not be able to sign on. You can determine if this has happened to you by looking in the statistics file for a signon record with an SAFA020I error message that matches your user ID. For more information on this error, refer to *Signon and IUI/API Errors* in *Connect:Direct for z/OS Administration Guide*.

Note: The main rationale behind the timeout feature is to do “housekeeping” in the system to keep the number of hung sessions to a minimum and avoid the limit specifying the maximum number of users that can be signed on to Connect:Direct. Without the timeout feature, an administrator would have to recycle the DTF to allow TCP/API logons to reconnect once the MAXUSER limit was reached.

Using SIGNON through the IUI

To execute the SIGNON command through the IUI, perform the following steps:

1. Select the Connect:Direct option from your ISPF/PDF Primary Option Menu. The Copyright screen and the Signon screen are displayed, unless SIGNON defaults are previously set.

```

CMD ==>

                                Connect:Direct for z/OS
                                Sterling Commerce, Inc.

                                VERSION          : VV
                                RELEASE          : RRR
                                MODIFICATION     : MMM
                                LEVEL           : PPPP

                                TODAY IS       : yyyy/mm/dd (yyyy.ddd )           THE TIME IS: hh:mm

USER ID =>
PASSWORD =>
NEW PASS =>
NODE NAME => _____
TRANSPORT => _____
COMMUNICATION ADDRESS =>( 12345 , _____ )
DESCRIPTION => _____
TIMEOUT => ____

ALL COMMANDS FOR THIS SESSION ARE TO BE CASE SENSITIVE ==> YES

```

The Signon screen indicates the version, release, modification, and Program Update Tape (PUT) levels of the Connect:Direct product. Sterling Commerce updates these numeric levels as it releases new versions of the Connect:Direct for z/OS product.

2. Type your user ID and press **Enter**.

Depending upon your site configuration, you may also need to type in a password, a new password, node name, transport, communications address, and case sensitivity option before pressing **Enter**. These parameters are described on page 61, or press **PF1** for Help.

If you are using the Extended Submit Facility (ESF), the DTF does not need to be active to execute the SIGNON, SIGNOFF, and SUBMIT commands.

Note: If the *Error requesting session, Connect:Direct may be inactive* message is displayed, the Connect:Direct DTF is not started.

If your signon is successful, the Primary Options Menu is displayed, unless you have updated your SIGNON defaults to execute an initial command.

Viewing Your Current Signon Parameters

To see how your signon parameters defaults are currently set, select the **WHO** option from the Primary Options Menu. The following figure shows an example of the User Information display produced by the WHO option.

```

                                Connect:Direct IUI USER INFORMATION                                hh:mm
CMD ==>

    USER ID==> USER01

        USER NODE           ==> node.name
        TRANSPORT PROTOCOL  ==> NETMAP
        COMMUNICATION ADDRESS ==> APPLID
        TEMPORARY DSN DDNAME ==> TMPDD
        TEMPORARY DSN VOLSER ==> TMPVSER

    DATA SETS:

        TEMPORARY DSN        ==> CD.TEMP.DSN
        MESSAGE DSN          ==> CD.MSG.VER01
        NETWORK MAP DSN      ==> CD.NETMAP.VER01
        PUBLIC PROCESS DSN   ==> CD.PUBLIB

        IUI VERSION          ==> VV
        IUI RELEASE          ==> RRR
        IUI MODIFICATION     ==> MMM
        IUI PUT LEVEL        ==> PPPP

```

Setting Up Signon Defaults

Use the Signon Defaults screen to set up signon default information. Select **SD** from the Primary Options Menu to access the Signon Defaults screen.

```

node.name                                SIGNON DEFAULTS                                hh:mm
CMD==>

CURRENT DEFAULTS:
USER ID ==>
PASSWORD ==>
TRANSPORT==>
COMMUNICATION ADDRESS ==>(           ,           )

DESCRIPTION ==>
TIMEOUT      ==> YES

TEMPORARY DATA SET NAME ==>
ALLOCATION TYPE           ==>
PRIMARY SPACE            ==>           UNIT TYPE ==>
SECONDARY SPACE          ==>           VOL=SER  ==>
EXTENDED SUBMIT FEATURE ==>
UPPER CASE FOR PRINT    ==>
PRINT FILE DESTINATION  ==>

PNODE ACCOUNTING DATA  ==>

SNODE ACCOUNTING DATA  ==>

DO YOU WANT ALL COMMANDS FOR THIS SESSION TO BE CASE SENSITIVE? ==> NO
INITIAL COMMAND         ==>

```

Note: Instead of users individually altering their signon defaults, the Connect:Direct administrator can define global signon defaults. With this method, users will not be responsible for increasing their allocation and can also avoid SB37 ABENDs. For more information, see *Global Signon Defaults* in the *Connect:Direct for z/OS Administration Guide*.

The INITIAL COMMAND field enables you to bypass the Connect:Direct Primary Menu and have a designated command screen display automatically at signon. For example, to have the User Information screen (WHO) display automatically at signon, type **WHO** in the Initial Command field.

Command stacking is allowed. For example, type **ADMIN.ST** in the INITIAL COMMAND field to branch to the View Type Record (ST) option of the Administration Menu (ADMIN).

By typing your user ID and other information in the SIGNON Defaults panel, you can bypass the SIGNON panel.

Note: You can also bypass the Signon screen if you are running a Stage 1 Signon exit, DMCXSIGN, and if you use the panel SGNONPAN provided in \$CD.SAMPLIB. Complete instructions are included in the SGNONPAN member.

Using SIGNON through the Batch Interface

The SIGNON command has the following format and associated parameters. The required parameters and keywords are in bold print. Default values for parameters and subparameters are underlined.

Label	Command	Parameters
(optional)	SIGNON	NETMAP = network map data set name CASE= Yes <u>No</u> COMADDR= (Port number, IP address alias name) ESF= <u>Yes</u> No FOLD= Yes <u>No</u> NODE= node name PACCT= 'pnode accounting data' PRINT= destination of printed output RECONNECT SACCT= 'snode accounting data' SPACE= (<u>CYL</u> TRK blk,([prim],[sec])) TMPDD= preallocated data set ddname TMPDSN= preallocated data set name TRANSPORT= SNA TCP <u>NETmap</u> UNIT= temporary dsn unit type USERID= (ID, pswd, newpswd) VOLSER= volume serial DESCRIPTION= '30-character string' TIMEOUT= <u>Yes</u> No

The following table describes each parameter.

Parameter	Description
NETMAP = network map data set name	<p>Specifies the network map data set name, which is a 1-44 alphanumeric character string with the first character alphabetic. This parameter is required. Ask your Connect:Direct administrator for information about completing this value.</p> <p>If you provide the DMNETMAP DD name in the job stream used with DMBATCH, you are not required to include this parameter with the SIGNON command. If you specify the data set name in both places, the network map data set name you specify on the SIGNON command takes precedence.</p> <p>For the IUI, the network map data set name is specified in the ISPF/PDF Primary Option Menu as a parameter of the DMISTART program. This data set name is established by your Connect:Direct administrator at installation and is not indicated through the IUI SIGNON command screens.</p>
CASE = Yes <u>No</u>	<p>The CASE parameter determines case sensitivity. It specifies whether parameters associated with accounting data, user ID, password, and data set name in the commands which follow the signon must be case sensitive. The default is No. This parameter is optional.</p> <p>You can override this designation on a command-by-command basis by adjusting the case sensitivity parameter on the individual command.</p>
COMADDR = (port number, IP address alias name)	<p>Specifies the communications address used in TCP/IP connections to the DTF. This parameter is specified only if the TRANSPORT parameter is set to TCP.</p> <p>The port number specifies the port number used for TCP/IP communications. This value is the value from the TCP.API.LISTEN initialization parameter of the DTF that you want to sign on to.</p> <p>The IP address/alias name specifies the IP address or alias name used for TCP/IP communications.</p> <p>If you use the COMADDR parameter, you must specify both the port number and IP address/alias name.</p> <p>If you use alias name or long domain name resolution, you do not need to supply the IP address in the COMADDR parameter or in the COMMUNICATIONS ADDRESS field during Connect:Direct SIGNON. The name resolution feature resolves a valid alias name or long domain name to the matching IP address. See <i>Signing On to the API</i> on page 66 for more information.</p> <p>The maximum length of the IP address alias name variable is 16 characters. If you want to use an LDNS (long domain name system) or an alias name greater than 16 characters, the name must be defined in the network map, and the TRANSPORT must be coded as NET.</p> <p>Note: Allocate the SYSTCPD DD to the TCPDATA file in both the Connect:Direct startup JCL and API session startup JCL when you request domain name resolution.</p>

Parameter	Description
ESF = <u>Yes</u> No	<p>Specifies whether the Extended Submit Facility (ESF) is available for the current signon. This parameter is optional. The default value is Yes.</p> <p>The ESF enables you to submit Processes even if the DTF is not active. When signing on to a Connect:Direct DTF that is active, but with no VTAM APPLID available, the ESF enables signon to the DTF. When you submit the Process it is enabled and placed in the TCQ as soon as the ESF.WAIT time expires.</p> <p>ESF is available only for Processes submitted on the local node. You can only issue the Connect:Direct SIGNON, SUBMIT, and SIGNOFF commands using the ESF; all other Connect:Direct commands are rejected.</p>
FOLD = Yes <u>No</u>	<p>Determines printing upper case control. It specifies whether printed output, an option of the various SELECT commands, are all uppercase letters. The default value is No, specifying printed output of uppercase and lowercase letters. This parameter is optional.</p>
NODE = node name	<p>Specifies the name of the node (DTF) you sign on to. It is the name assigned to a node defined in the network map definitions. This parameter is optional.</p>
PACCT = 'pnode accounting data'	<p>Specifies the accounting data for the primary node (PNODE). The maximum length of the accounting data is 256 characters. If special characters, such as a space, are part of the accounting data, you must enclose the string in single quotation marks. Connect:Direct uses this data as a default for each Process unless you override it on the SUBMIT command, PROCESS statement, or SUBMIT statement. This parameter is optional.</p>
PRINT = print spool file destination	<p>Specifies an allocation string that determines the destination of the printed output that you request on the various SELECT commands. This parameter is optional. SYSOUT=A is the default value. An example follows.</p> <p>'SYSOUT=C DEST=RMT1 FREE=CLOSE RETURN=(DD)'</p> <p>You must include both FREE=CLOSE and RETURN=(DD) in the string for this specification. When using the IUI, if you do not specify a destination, Connect:Direct routes the output to the SYSOUT class specified in the initialization parameters (external to Connect:Direct) under the TSO user ID. If you modify the destination, you need to sign off Connect:Direct after updating the field in order for the new destination to be in effect.</p>
RECONNECT	<p>Specifies that you can reestablish your VTAM session without having to sign off and sign on again. You can lose a VTAM session due to a link or line failure. This parameter is optional. The USERID parameter is the only other parameter that you specify with the RECONNECT option.</p>
SACCT = 'snode accounting data'	<p>Specifies the accounting data for the secondary node (SNODE). This parameter is optional. The maximum length of the accounting data is 256 characters. If special characters, such as a space, are part of the accounting data, you must enclose the string in single quotation marks. Connect:Direct uses this data as a default for each Process unless you override it on the SUBMIT command, PROCESS statement, or SUBMIT statement.</p>

Parameter	Description
SPACE = (allocation type, ([prim], [sec]))	<p>Specifies the temporary data set (TMPDSN) allocation type, primary space allocation, and secondary space allocation. This parameter is optional.</p> <p>The first subparameter is the allocation type. An allocation type of CYL specifies space allocation by cylinder. CYL is the default value. An allocation type of TRK specifies space allocation by track. An allocation type of blk specifies space allocation by the average block length of the data. Connect:Direct computes the number of tracks to allocate.</p> <p>The second subparameter, prim, specifies the primary allocation of storage. The default value is 1.</p> <p>The third subparameter, sec, specifies the secondary allocation of storage. The default value is 1.</p>
TMPDD = preallocated data set ddname	<p>Specifies the ddname of a user-supplied preallocated data set. The maximum length is an 8-character alphanumeric string. This parameter is optional. Do not use both the TMPDD and TMPDSN parameters at the same time in the SIGNON command.</p>
TMPDSN = preallocated data set name	<p>Specifies the name of a preallocated data set used to receive command results that are printed or displayed. The data set name is a 1-44 alphanumeric character string that must conform to z/OS standards for data set names. This parameter is optional, and when not specified, Connect:Direct builds a system temporary data set. It normally derives the name for the data set based on your TSO ID. In some cases, this derived data set name violates your facility naming convention. Use the Signon Defaults screen to override the data set name as needed.</p> <p>If you name a data set at signon time, preallocate it with the following attributes prior to signing on to Connect:Direct (4104 is the minimum BLKSIZE): DSORG=PS LRECL=4100 RECFM=VBA BLKSIZE=4104</p> <p>Do not use both the TMPDD and TMPDSN parameters at the same time in the SIGNON command.</p>
TRANSPORT = SNA TCP <u>NETmap</u>	<p>Defines the communications protocol to use for the session. This parameter is optional.</p> <p>The default value (NETmap) indicates that the protocol defined in the network map is used for the session.</p> <p>A value of SNA tells Connect:Direct to establish the session using SNA. You must define the node you are signing on to as an SNA node, or the SIGNON Process fails.</p> <p>A value of TCP tells Connect:Direct to establish a TCP/IP session. Connect:Direct uses the communications address specified in the COMADDR parameter (see page 62 for a description of the COMADDR parameter), if that parameter is specified as part of the SIGNON command. If the COMADDR parameter is not specified, Connect:Direct tries to establish the session using the communications address from the network map. If an invalid COMADDR parameter is supplied, Connect:Direct establishes an ESF API session. See <i>Signing On to the API</i> on page 66 for more information.</p>
UNIT = temporary dsn unit type	<p>Specifies the temporary data set name (TMPDSN) unit type, which must be a 1-8 alphanumeric character string defining the unit type for data set allocation. This parameter is optional. If omitted, the operating system defined default for DASD is used by dynamic allocation.</p>

Parameter	Description
USERID = (ID, pswd, newpswd)	<p>Specifies your signon security information. This parameter is optional.</p> <p>ID specifies the security ID that Connect:Direct passes to a security exit. It can contain 1-64 characters of any kind, and is case-sensitive.</p> <p>pswd specifies the current security password. The security exit uses this parameter to validate the current security password. It contains 1-64 alphanumeric characters, and is case-sensitive.</p> <p>newpswd specifies the new security password. The security exit uses this parameter to change the current security password to the new security password. It contains 1-64 alphanumeric characters, and is case-sensitive.</p> <p>When you attempt a sign on to a remote node, the user ID and password information must correspond to security procedures at that remote node.</p>
VOLSER = volume serial	<p>Specifies the temporary data set name (TMPDSN) volume serial number, which must be a 1-8 character alphanumeric string defining the volume serial number for data set allocation. This parameter is optional.</p>
DESCRIPTION = 'optional 30-character string'	<p>Describes the API type that is logging on. It is displayed with the Select Task command. If Connect:Direct cannot determine the API type, the communication address is used.</p> <p>Defaults: DMCHLAPI, DMBATCH, ISPFUI, CONSOLEUI, ICOUI, CICSUI, and Java Application Interface</p>
TIMEOUT = <u>YES</u> NO	<p>Specifies whether the value specified for the TCP.API.TIMER global initialization parameter applies to this specific user. This parameter is optional. The system will only accept changes to this field made by an administrator and only for their own IUI session; all others will be governed by the initialization parameter. The default value is Yes. Do not use the TIMEOUT keyword in DMBATCH.</p> <p>For more information, see <i>Timeout Problems</i> on page 57. Additional information about the TCP.API.TIMER parameter can be found in Appendix A, <i>Global Initialization Parameters</i>, in the <i>Connect:Direct for z/OS Administration Guide</i>.</p>

To use the SIGNON command from the batch interface, perform the following steps:

1. Place the command in the DMBATCH job stream as described in Chapter 3, *The Batch Interface*.
2. Submit the job while Connect:Direct is running. If you are using the Extended Submit Facility (ESF), the DTF does not need to be active for the SIGNON command to process.

The following example shows a SIGNON command in which the user specifies a preallocated temporary data set named MYUID1.TEMP.DSN which resides on SYSDA. The accounting data is specified for the PNODE and SNODE.

```
SIGNON    NETMAP=DALLAS.NETMAP  -
USERID=(MYUID1,MYPSWD)  -
PACCT='JOB FOR SYSMANT, DEPT.27, MARKETING'  -
SACCT='JOB FOR INVENTORY, DEPT.55, ACCOUNTING'  -
TMPDSN='MYUID1.TEMP.DSN'  -
UNIT=SYSDA
```

See page 66 for additional examples of the SIGNON command in the batch interface.

Signing On to the API

The communication address used to establish a connection to Connect:Direct is determined by the TRANSPORT parameter defined in the SIGNON command. The default for the TRANSPORT parameter is NET (NETMAP) which means that the protocol defined in the NETMAP Adjacent node entry is used. See *Adjacent Node Definition Examples* on page 66 .

Parameter Value	Description
TRANSPORT = NET	Default. When TRANSPORT is defined as NET, the sign on process retrieves the Adjacent node entry to determine if the TCPAPI parameter has been defined. If TCPAPI exists, then a TCP connection is attempted using the communication address defined. The communication port number is obtained from the TCPAPI parameter, and if the IP address exists in the TCPAPI parameter, it is also used. If the IP address does not exist in the TCPAPI parameter, it must be obtained from either the Adjacent node or the LDNS parameter. If the TCPAPI does not exist, the APPLID parameter is retrieved and SNA is used as the protocol.
TRANSPORT = SNA	When TRANSPORT is defined as SNA, the sign on process retrieves the Adjacent node entry to determine the APPLID parameter, and SNA is used as the protocol. If the APPLID parameter does not exist, then an ESF SIGNON is performed.
TRANSPORT = TCP	When the TRANSPORT is defined as TCP, the communication address must be specified on the SIGNON command. To initialize Connect:Direct without SNA support, refer to <i>Initializing Connect:Direct without SNA Support</i> in the <i>Connect:Direct for z/OS Administration Guide</i> .

Adjacent Node Definition Examples

To only allow SNA API signons:

```
/* PNODE=SNODE WITH SNA API SIGNON ONLY */
ADJACENT.NODE=(( CD.OS390.NODE,M1DEV93C) PARSESS=(53 2) -
APPLIDS=(M1CDI701 M1CDI702 M1CDI703) -
)
```

To only allow TCP API signons using an IPv4 address:

```
/* PNODE=SNODE WITH TCP API SIGNON ONLY USING IP ADDRESS */
ADJACENT.NODE=(( CD.OS390.NODE,M1DEV93C) PARSESS=(53 2) -
TCPAPI=(4198,111.222.333.444) -
)
```

To only allow TCP API signons using an IPv6 address:

```
/* PNODE=SNODE WITH TCP API SIGNON ONLY USING IP ADDRESS */
ADJACENT.NODE=(( CD.OS390.NODE,M1DEV93C) PARSESS=(53 2) -
TCPAPI=(4198,1111:2222:3333:4444:5555:6666:7777:8888) -
)
```

To only allow TCP API signons using the LDNS parameter:

```

/* PNODE=SNODE WITH TCP API SIGNON ONLY USING LDNS */
ADJACENT.NODE=(( CD.OS390.NODE,M1DEV93C) PARSESS=(53 2) -
TCPAPI=(4198,) -
LDNS=long.domain.name -
)

```

To allow both SNA and TCP API signons:

```

/* PNODE=SNODE WITH BOTH SNA AND TCP API SIGNON */
ADJACENT.NODE=(( CD.OS390.NODE,M1DEV93C) PARSESS=(53 2) -
TCPAPI=(4198,111.222.333.444) -
APPLIDS=(M1CDI701 M1CDI702 M1CDI703) -
)

or

/* PNODE=SNODE WITH BOTH SNA AND TCP API SIGNON */
ADJACENT.NODE=(( CD.OS390.NODE,M1DEV93C) PARSESS=(53 2) -
TCPAPI=(4198,) -
LDNS=long.domain.name -
APPLIDS=(M1CDI701 M1CDI702 M1CDI703) -
)

```

Signing On to Multiple Sessions

Use the SIGNON command to sign on simultaneously to multiple sessions of the DTF. The SIGNON command parameters which are valid for the multiple session signon are a subset of the full SIGNON command parameters. See *Using SIGNON through the Batch Interface* on page 61 for descriptions of these parameters.

Master Session

The first successful SIGNON command establishes the master session. The master session is the active session until another session is established. Each subsequent signon becomes the active session unless sessions are swapped using the SWAP NODE command.

Signing on to the master session and signing on to another node under the master session requires two different signons. The parameters specified for the master session signon dictate the environment for all sessions running under the master session.

Non-Master Session

For a session other than master, you are limited to the following SIGNON command parameters.

Label	Command	Parameters
(optional)	SIGNON	NODE= node name
		RECONNECT
		USERID= (ID, pswd, newpswd)

See page 61 for descriptions of these parameters.

SIGNOFF Sequence

Carefully control your SIGNOFF command sequence because if you sign off the master session, Connect:Direct signs off all other sessions. If you sign off a non-master session, Connect:Direct only signs off that session, and the master session immediately becomes the active session again, regardless of the order in which the sessions are established. See *Sequencing the SIGNON, SWAP NODE, and SIGNOFF Commands* on page 72 for examples of the importance of the sequence of your SIGNOFF commands.

Switching Nodes

When you issue multiple signons, one node can perform one function and another node a different function. The SWAP NODE command enables you to swap from node to node. See page 70 for the SWAP NODE command description and examples.

Using the Batch Interface for Multiple Session SIGNON

To use the multiple session SIGNON command through the batch interface, place your commands in the DMBATCH job stream and submit the job while Connect:Direct is running. If you have the Extended Submit Facility (ESF) available, the DTF does not need to be running to submit the job.

See the examples beginning on page 72 for job streams that issue SIGNON, SWAP NODE, and SIGNOFF commands and an explanation of the significance of the command sequence in the job stream.

Using the IUI Multiple Session SIGNON Command

To access the Multiple Session Signon screen, select **MS** from the Primary Options Menu.

```

node.name                                hh:mm
CMD ==>                                  yyyy.mm.dd
                                           yyyy.ddd

                                MULTIPLE SESSION SIGNON

USER ID  ==> USER01
PASSWORD ==>
NEW PASS ==>
NODE NAME ==>
TRANSPORT ==> NET
COMMUNICATION ADDRESS ==> (           ,           )
DESCRIPTION ==>
RECONNECT ==> N   (Y OR N. TO THE *ACTIVE* NODE
                   AFTER A SESSION FAILURE)
PNODE ACCOUNTING DATA ==>
SNODE ACCOUNTING DATA ==>

DO YOU WANT ALL COMMANDS FOR THIS SESSION TO BE CASE SENSITIVE? ==> NO

```

The Multiple Session Signon screen enables you to sign on to more than one node during a single ISPF/PDF session. The APPLIDS value in the network map Adjacent Node definition determines the maximum number of sessions that Connect:Direct is allowed on one DTF. Each signon uses an additional VTAM APPLID.

See page 61 for field descriptions, or press the **PF1** key for Help.

Swapping Nodes

The SWAP NODE command enables you to swap to another node when you are signed on to more than one node at a time. Use the SWAP NODE command to specify which node a command is issued to.

Using SWAP NODE through the UI

You can use the Swap/Display User Sessions screen to swap to another node or delete (signoff) another node.

1. Select option **SW** from the Primary Options Menu to access the SWAP/DISPLAY User Sessions screen. The screen displays all user sessions for this ISPF/PDF session.

-----SWAP/DISPLAY USER SESSIONS-----				Row 1 to 2 of 2
==>				SCROLL ==> PAGE
NODE	COMMUNICATION ID	CURRENT	STATUS	
	USERID			

node.name1		applid0	*ACTIVE*	*MASTER* **LOCAL*
	userid2			
node.name3		applid1		
	userid1			
***** Bottom of data *****				

2. Do one of the following:
 - ◆ To swap sessions, type **S** in the input field to the left of the node name and press **Enter**. ACTIVE is displayed next to the node name you just selected. To determine which node is currently signed on when this field is not displayed, look to the upper left corner of most panels for the display of the node name.
 - ◆ To delete or sign off a session, type **D** in the input field to the left of the node name and press **Enter**.

Note: If you sign off the master session (the first node you signed on to in this session), all sessions are lost.

Using SWAP NODE through the Batch Interface

The SWAP NODE command has the following format and associated parameters.

Label	Command	Parameters
(optional)	SWAP NODE	node name

The only parameter for the SWAP NODE command is node name, which specifies which node to swap to in a multiple session environment. The node name is the name assigned to the LOCAL.NODE field defined in the network map definitions.

To use the SWAP NODE command from the batch interface:

1. Place your commands in the DMBATCH job stream as described in Chapter 3, *The Batch Interface*.
2. Submit the job while Connect:Direct is running.
3. Verify your results.

The following command swaps you from your current active remote node with a user ID of JONES2 to the local node under the user ID of JONES1.

```
//SYSIN DD *
SIGNON USERID = (JONES1) -
        NODE = CD.LOCAL
SIGNON USERID = (JONES2) -
        NODE = CD.REMOTE
SWAP NODE CD.LOCAL
SELECT NETMAP WHERE (NODE=CD.LOCAL)
SIGNOFF
SIGNOFF
/*
```

Signing Off of Connect:Direct

The SIGNOFF command terminates a connection to the DTF. If you are using the Extended Submit Facility (ESF), you can submit the SIGNOFF command even though the DTF is inactive.

Using SIGNOFF through the IUI

The following table identifies how to SIGNOFF from various locations of the IUI. You must press **Enter** after each SIGNOFF command.

SIGNOFF Command	Location
SIGNOFF	SIGNON screen or the Primary Options Menu
End	SIGNON screen or the Primary Options Menu
X	SIGNON screen or the Primary Options Menu
=SIGNOFF or =X	Command Line

When you sign off Connect:Direct, the system returns you to the ISPF menu.

Follow these rules when you sign off Connect:Direct:

- ◆ You must type a SIGNOFF command twice from the command line.

Note: The first SIGNOFF command returns you to the Primary Options menu and the second SIGNOFF command returns you to the ISPF menu.

- ◆ You only need to sign off once from the SIGNON screen or the Primary Options Menu.

Using SIGNOFF through the Batch Interface

The SIGNOFF command has the following format. No associated parameters are available.

Label	Command	Parameters
(optional)	SIGNOFF	

See page 67 for information about the significance of the SIGNOFF command sequence, if you are using the Multiple Session Signon feature.

To use the SIGNOFF command from the batch interface, perform the following steps:

1. Place your commands in the DMBATCH job stream as described in Chapter 3, *The Batch Interface*.
2. Submit the job while Connect:Direct is running. If you have the Extended Submit Facility (ESF) available, the DTF does not need to be running in order for the SIGNOFF command to function.
3. Verify your results.

The following example shows a SIGNON command followed by a SIGNOFF command.

```
SIGNON  USERID=(MYUID1,MYPSWD) NETMAP=NETMAP.NAME
SIGNOFF
```

See the examples in *Sequencing the SIGNON, SWAP NODE, and SIGNOFF Commands* on page 72 for more examples of the SIGNOFF command used with the Multiple Session Signon feature and the SWAP command.

Sequencing the SIGNON, SWAP NODE, and SIGNOFF Commands

The importance of the sequence in which you issue your SIGNON, SWAP NODE, and SIGNOFF commands is illustrated by the following examples.

Example 1: Terminating Two Sessions with One SIGNOFF Command

The following example shows a DMBATCH SIGNON to two different nodes and the use of the SWAP NODE command. The SIGNOFF command in this example causes Connect:Direct to terminate both sessions, because the active session at signoff is the master session.

```
/* NETMAP = CD.NETMAP, BOSTON IS ACTIVE (MASTER) */
SIGNON  USERID=(SAM) NODE=CD.BOSTON
/* SIGN ON TO JERSEY */
SIGNON  USERID=(SAM) NODE=CD.JERSEY
/* SUBMIT COPY ON JERSEY */
SUB  PROC=COPY1
/* SWAP TO BOSTON */
SWAP NODE  CD.BOSTON
/* SUBMIT COPY ON BOSTON */
SUB  PROC=COPY1
/* SIGN OFF BOTH NODES (terminating master session)*/
SIGNOFF
```

Example 2: Terminating a Master Session with the SIGNOFF Command

In the following DMBATCH example, the first SIGNOFF command terminates the CD.NEWYORK node because it is the last SIGNON command issued. The second SIGNOFF command terminates both the CD.JERSEY and CD.BOSTON nodes because the first SIGNOFF command made the master session the active session and signing off the master session signs off all other sessions.


```

/* NETMAP = CD.NETMAP, BOSTON IS ACTIVE (MASTER */
SIGNON  USERID=(SAM) NODE=CD.BOSTON
/* SIGN ON TO JERSEY */
SIGNON  USERID=(SAM) NODE=CD.JERSEY
/* SIGN ON TO NEWYORK */
SIGNON  USERID=(SAM) NODE=CD.NEWYORK
/* BOSTON IS ACTIVE AFTER SIGNOFF TO CD.NEWYORK */
SIGNOFF
/* SIGN OFF ALL SESSIONS (terminating master session) */
SIGNOFF

```

Example 3: Terminating a Non-Master Session with the SIGNOFF Command

The following example illustrates that whenever a SIGNOFF command is issued to a non-master session, the master session automatically becomes active again regardless of the order in which the nodes are signed on.

If you try to swap to a node that is already active, Connect:Direct ignores the SWAP NODE command. The USERID (or any other user ID specified) must be a user record in the Connect:Direct authorization file or a valid ID for the security subsystem in use, to sign on to the node.

```

/* NETMAP = CD.NETMAP, BOSTON IS ACTIVE (MASTER) */
SIGNON  USERID=(SAM) NODE=CD.BOSTON
/* SIGN ON TO JERSEY */
SIGNON  USERID=(SAM) NODE=CD.JERSEY
/* SIGN ON TO NEWYORK */
SIGNON  USERID=(SAM) NODE=CD.NEWYORK
/* NEWYORK ACTIVE */
SIGNOFF
/* SIGN ON TO JERSEY */
SWAP NODE  CD.JERSEY
/* JERSEY ACTIVE */
SIGNOFF
/* SIGN OFF ALL SESSIONS */
SIGNOFF

```

Building, Modifying, and Submitting Processes

This chapter provides the information you need to know to submit Processes, including the syntax, format, and instructions for the IUI and batch interface. This chapter includes the following topics:

- ◆ Overview of Process Execution
- ◆ The SUBMIT command

Chapter 8, *Controlling Processes in the TCQ*, describes how to monitor and manipulate your submitted Processes.

Note: The maximum size allowed for a Process is 1 MB.

Note: You can also use the Connect:Direct Browser to perform some of the procedures in this chapter. To learn more about the Connect:Direct Browser, see the user's guide on the Connect:Direct Browser CD-ROM or available online from the Sterling Commerce Documentation Library.

Overview of Process Execution

This section provides an overview of how Connect:Direct executes submitted Processes. A series of examples immediately follows the overview to illustrate how Connect:Direct executes submitted Processes. You can construct and submit Processes for execution in several ways, depending upon the user interfaces available on your operating environment. These Connect:Direct user interfaces include:

- ◆ Interactive User Interface (IUI)
- ◆ Connect:Direct Browser
- ◆ Connect:Direct File Agent
- ◆ Batch interface
- ◆ User-written programs issued through the Connect:Direct Application Program Interface (API)

The following steps describe how a Process executes:

1. You submit a Process.

You create and submit a new Process or submit a predefined Process from a Connect:Direct Process library.

2. The parser checks the syntax of the Process.
3. The Process is queued for execution and the submit message is issued.

If the Process passes syntax checking, it is placed in a work queue based on Process parameters, such as priority, class, and start time.

The Connect:Direct work queues are jointly referred to as the Transmission Control Queue (TCQ) or the Process queue. A Process can have one of the following queue states in the TCQ:

State	Description
EXECUTION	Indicates that the Process is executing.
WAIT	Indicates that the Process is waiting until a connection is established or available.
HOLD	Indicates that the Process is held on the queue until released by an operator or the SNODE connects with a request for held work.
TIMER	Indicates that the Process is submitted to execute at the user-specified time or date.

You can manipulate any Process in the TCQ with various Connect:Direct commands that are described in the next chapter.

A message indicating that the Process submitted successfully is returned when the Process is placed in the TCQ.

4. Connect:Direct finds an available connection and Process execution begins. The Process is selected for execution based on Process parameters and the availability of the remote node.

Process Routing

The RETAIN, HOLD, and STARTT parameters route Processes as described in the following table.

Parameters	Queue	Comments
None	Wait	Process remains on the Wait queue until Connect:Direct can start a session with the SNODE at which time it moves to the Execution queue.
RETAIN=INITIAL	Hold	Process automatically executes each time Connect:Direct is initialized with TCQ=WARM.

Parameters	Queue	Comments
RETAIN=YES	Hold	A copy of the Process is kept in the Hold queue after it has executed. The Process does not execute again until it is released by a CHANGE PROCESS command.
HOLD=YES	Hold	Process remains in the Hold queue until someone releases the Process.
HOLD=CALL	Hold	Process is automatically moved from the Hold queue to the Wait queue when the SNODE contacts the node on which the Process resides.
STARTT	Timer	When the scheduled time and date arrive, the Process is put on the wait queue and is available for execution.

RETAIN=INITIAL is useful for Processes that contact other Connect:Direct nodes each time Connect:Direct completes initialization. This action causes any work queued on the remote node for the local node to begin. The HARTBEAT sample Process allows you to test if Connect:Direct is running on a SNODE without actually running a Process on the node. For more information, see the note on the HARTBEAT Process in the *Maintaining the Network* chapter in the *Connect:Direct for z/OS Administration Guide*.

You can use RETAIN=YES combined with STARTT to run a Process at a periodic interval. For example, RETAIN=YES and STARTT=(Tuesday, 3pm) starts the Process every Tuesday at 3 pm; RETAIN=YES and STARTT=(,12:00) starts the Process each day at noon.

The Timer queue is also used for session retry and file allocation retry based on Connect:Direct initialization parameters specified by a particular installation. When retry limits are exhausted, the Process is moved to the Hold queue with a WC (hold queue, waiting connection) status for session retry.

Note: For allocation failures to be requeued, you must set the REQUEUE initialization parameter to YES so that the Process is placed in the HO RA (HO = Held by Operator, RA = Held for Restart Due to Allocation Error) queue.

Process Queuing

When you submit a Process, Connect:Direct puts the Process in a logical queue based on the Process statement routing parameters on page 76.

Connect:Direct selects Processes in a first-in first-out manner for execution within Process class and priority as sessions are available. The following section describes in detail how Connect:Direct selects Processes for execution.

Process Selection

Connect:Direct uses the parallel sessions capability of VTAM so that multiple Processes can execute simultaneously between any two Connect:Direct nodes. Process selection for each parallel session is based on a class that the user specifies on a Process.

Note: Parallel sessions support requires you to specify PARSESS=YES in the VTAM application definition for both nodes. If two nodes have differing values for parallel sessions, transfers are limited by the maximum number of sessions in the ADJACENT.NODE definition on the node where the Process is submitted.

You define the maximum number of sessions between two nodes in the network map. Because each session has a corresponding class value, the maximum number of sessions and maximum number of classes are equal. Selection of a Process for execution in a given node is based on Process priority (the PRTY parameter of the Process statement) within session class. User-specified class values allow a Process to execute on the session having the matching class value or on sessions with higher class values. The default class is the value specified in the Connect:Direct network map.

When one Process completes, Connect:Direct selects another Process (if any) to run on the available session. Connect:Direct searches until it finds the first Process with a class eligible to execute on the available session.

A typical use for classes is to define critical Processes with low class values so that more sessions are available for their execution. You can specify higher class numbers for time-consuming Processes; this enables sessions with corresponding lower class numbers to become available more frequently.

Intelligent Session Retry

The Intelligent Session Retry feature provides the ability to manage session retries when a connection to a specific node is down so that all Processes are not retried at the same time. This feature makes the existing connection retry facility more efficient by reducing the extra processing overhead created when each Connect:Direct Process retries the connection independently.

When multiple Processes are submitted for a specific node that is down, the Processes all initially attempt to connect. When this fails, the first Process submitted to this node continues to retry until MAXRETRIES is exceeded. The other Processes to this node are moved to the Hold (HO) queue with a RH (held for retry) status and are not retried. When MAXRETRIES is exceeded for the first Process, it is moved to the Hold (HO) queue with a WC (waiting connection) status for session retry.

After the first Process is moved to HO WC status, it can be restarted by one of the following ways:

- ◆ You can manually release the Process causing it to be retried until a connection is made or MAXRETRIES is exceeded, when it will return to HO WC status
- ◆ Another Process is submitted to this same node, causing the Process in HO WC status to be retried until either a connection is made or MAXRETRIES is exceeded, returning it to HO WC status
- ◆ The SNODE initiates communication with the PNODE, causing the Process in the HO WC status to be released

When the first Process in HO WC status connects and is released, this causes the other Processes being held in HO RH status to be released serially. Therefore, only one Process going to an inactive node will retry until the connection is made resulting in less overhead, since multiple Processes going to the same node are not all attempting to retry at the same time. This also reduces the number of statistic records which would be produced for multiple retries.

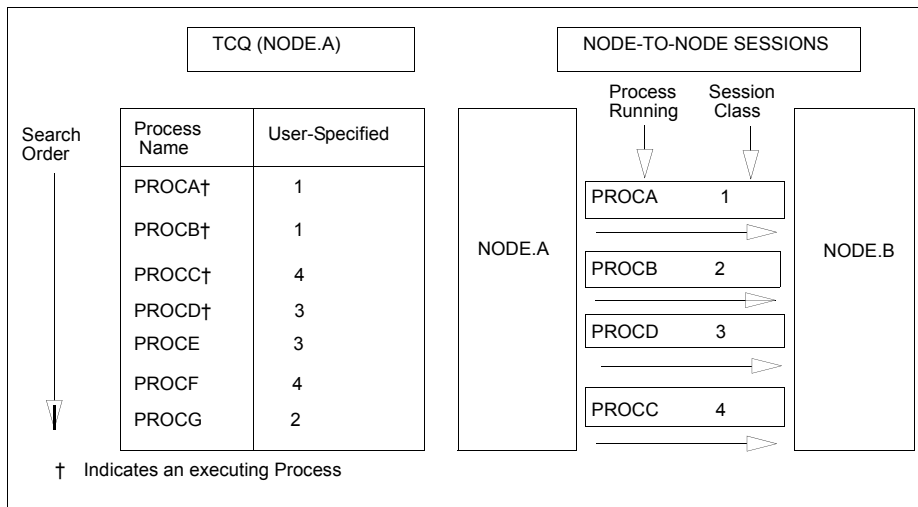
Note: The node name used by the Intelligent Retry feature is the specific name used for the SNODE parameter. This means that if the SNODE is known by more than one node name or TCPNAME value, only those in the queue waiting with that value will be recognized for release.

Process Execution Example

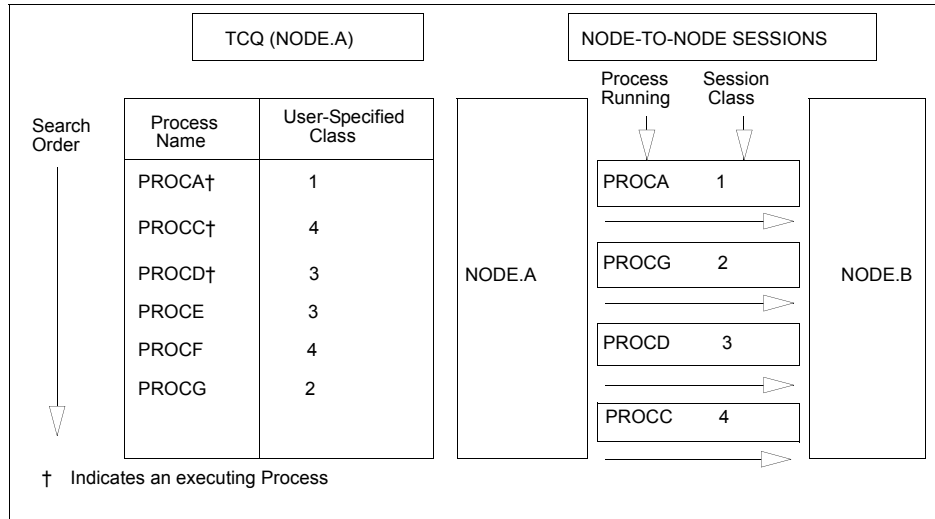
In the following example, Connect:Direct is up. Seven Processes are submitted to the TCQ on NODE.A to execute on NODE.B. All of the submitted Processes are ready to run. In addition, all Processes have a user-specified class value and the same priority. (Class determines which session Connect:Direct selects to run. Class values allow a Process to execute on a session with a matching class value or on sessions with higher class values.)

This site configuration enables up to four sessions to run between NODE.A and NODE.B. Each session between NODE.A and NODE.B has its own unique class number. Connect:Direct executes in the following order:

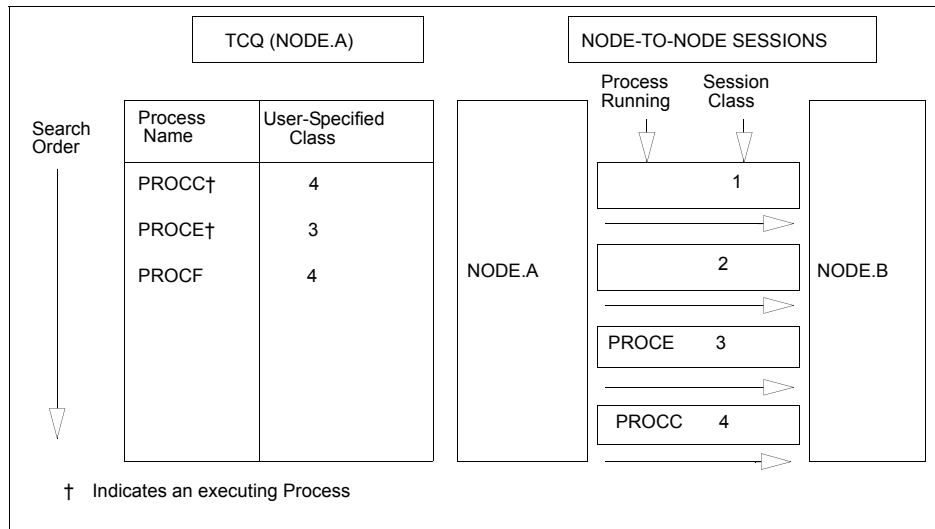
1. NODE.A simultaneously starts four sessions, as in the following figure. Processes are displayed in the same order they appear in the queue. Because the user-specified class of PROCA and PROCB is 1, they can run on the class 2, 3, or 4 sessions, if needed.



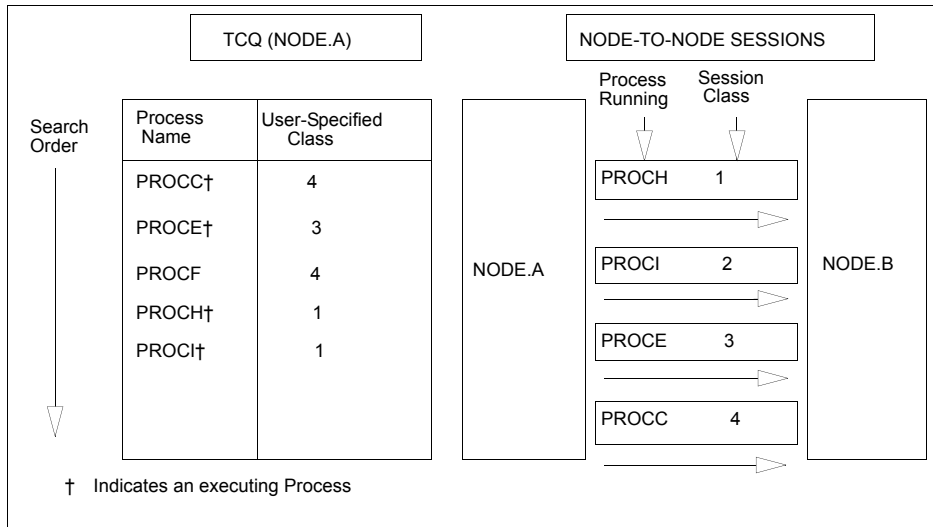
2. PROCB completes execution, making a session available. Connect:Direct looks through the TCQ for the first eligible Process for that session. PROCG is the next Process available to run on the class 2 session because all other Processes have a class value higher than 2.



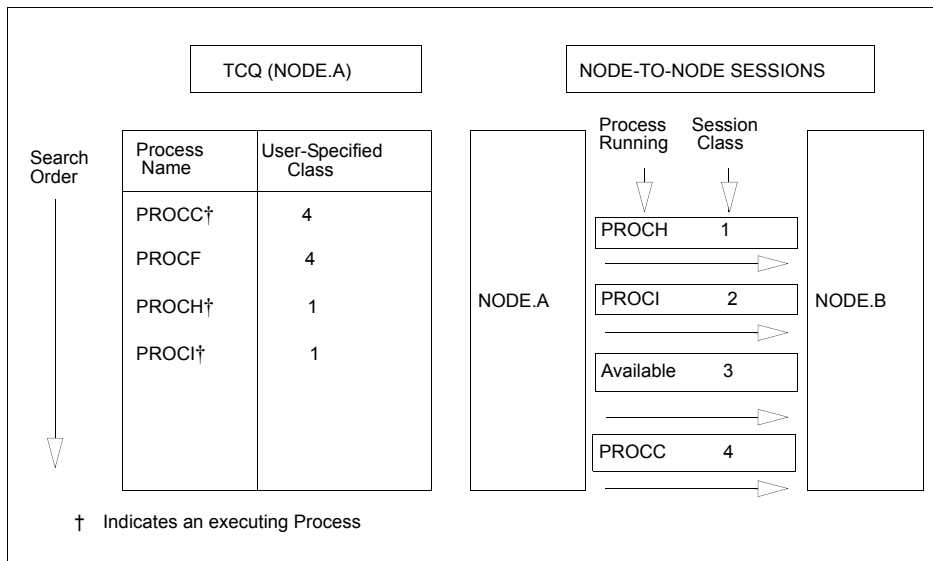
3. PROCA, PROCG, and PROCD complete execution. PROCE begins executing in the class 3 session and PROCC begins executing in the class 4 session. The class 1 and class 2 sessions cannot be used because the only Process remaining in the queue (PROCF) is class 4. PROCF must wait for an available class 4 session.



- PROCH and PROCI are submitted. PROCF continues to wait, because it can only run in a class 4 session.



- PROCE completes. PROCF is still waiting for PROCC to complete. If another Process is submitted for class 1, 2, or 3, it can use the available class 3 session.



Process Management in a Connect:Direct/Plex

In a Connect:Direct/Plex, the Connect:Direct Manager uses a set of Process management rules to schedule Processes to Connect:Direct Servers. These rules are applied in the following order:

- The Connect:Direct Manager first checks if the remote node is running a Connect:Direct release that supports multiple servers. If the remote node does not support multiple servers, the Connect:Direct Manager schedules all Processes to the Connect:Direct Server that is already running a Process with this node, and Process management does not occur. If the node

supports multiple servers, the Connect:Direct Manager continues with the Process management rules that follow.

2. The Connect:Direct Manager determines which Connect:Direct Servers on the remote node can run the Process, based on the PLEXCLASS of the Connect:Direct Server and the Process, and the current transport support in the Connect:Direct Server.
3. The Connect:Direct Manager then determines which Connect:Direct Server from step 2 has the lowest workload. The current Process workload of each Connect:Direct Server is determined by dividing the number of active Processes on the server by its MAXPROCESS initialization parameter value.
4. If two or more Connect:Direct Servers have the lowest workload, the Process is sent to the Connect:Direct Server that has the earliest last scheduled time (the time of day that the Connect:Direct Server Process is scheduled).

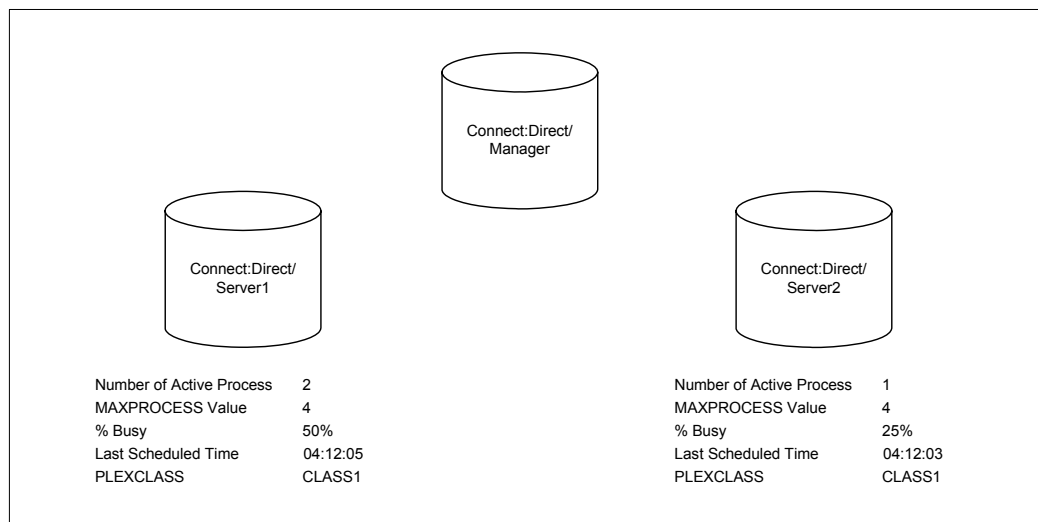
Note: If a Connect:Direct/Plex communicates with an external Connect:Direct system, the external system must send work to the Connect:Direct Manager in order to balance Process workloads in the Connect:Direct/Plex. If an external Connect:Direct system communicates directly with a Connect:Direct Server, work from the external system is not distributed using Process management; the Connect:Direct Server simply does the requested work. However, the Connect:Direct Server still informs the Connect:Direct Manager that it is processing work, so the Connect:Direct Manager can continue to correctly balance Process workload in the Connect:Direct/Plex.

Also, to perform workload balancing with an external Connect:Direct system when the Connect:Direct/Plex is the SNODE, the external system must be a Connect:Direct version that supports the Connect:Direct/Plex environment. You do not have to configure the external system as a Connect:Direct/Plex. Configure it to communicate directly with the Connect:Direct Manager.

Example

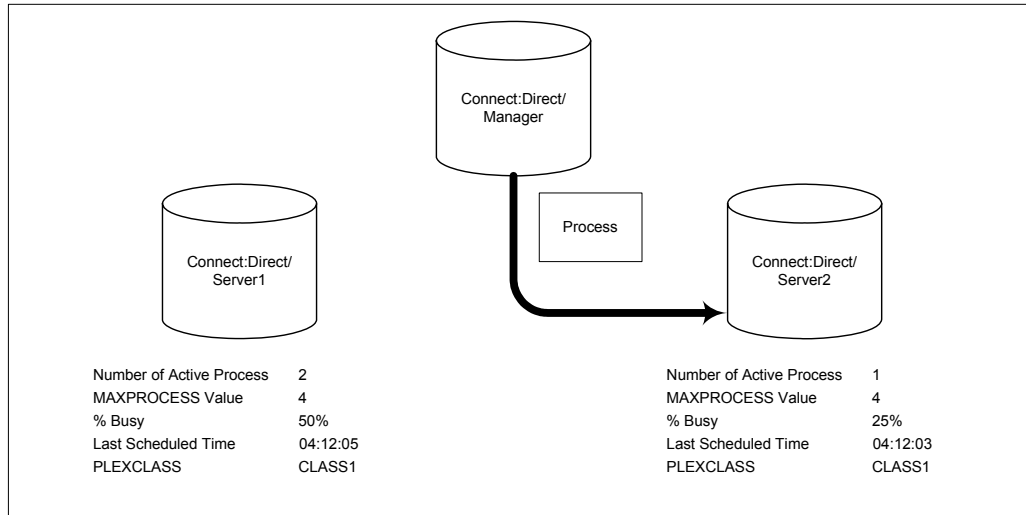
The following is an example of the steps in Process management.

1. A Connect:Direct/Plex consists of a Connect:Direct Manager and two Connect:Direct Servers. Both Connect:Direct Servers can process the same PLEXCLASS (CLASS1).

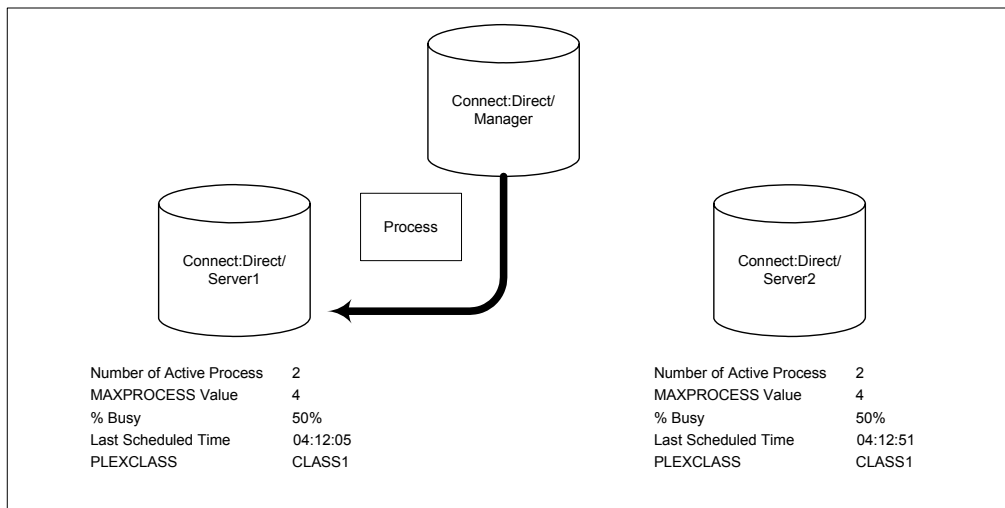


- The Connect:Direct Manager receives a new Process. After determining that both Connect:Direct Servers support the PLEXCLASS required to run the Process, the Connect:Direct Manager determines which Connect:Direct Server has the lowest workload.

In this case, SERVER2 has the lowest workload; it has one active Process out of a maximum of four. The Connect:Direct Manager routes the Process to SERVER2, and the last scheduled time of SERVER2 is updated to the current time.



- The Connect:Direct Manager has another Process to schedule. SERVER1 and SERVER2 have the same workload. However, the last Process is scheduled to SERVER2, so the Connect:Direct Manager routes the new Process to SERVER1.



The SUBMIT Command

The SUBMIT command enables you to submit a Process to the TCQ for execution. Parameters you specify on the SUBMIT command override any corresponding parameters specified in the Process itself.

Note: If you are using the Extended Submit Facility (ESF), you can use the SUBMIT command to submit Processes to the TCQ, even if the DTF is inactive.

The SUBMIT command has the following format. The Label is optional.

Label	SUBmit	Parameters and Subparameters
-------	--------	------------------------------

The SUBMIT command has the following parameters. The required parameters and keywords are in bold print. Default values for parameters and subparameters are underlined.

Label	Command	Parameters
(optional)	SUBmit	PROC= member or DSN= dsn dsn(member) CASE= Yes <u>No</u> CLASS= n DEBUG= trace settings for this Process HOLD= Yes <u>No</u> Call MAXDELAY= [<u>unlimited</u> queued 0 hh:mm:ss] NEWNAME= newname NOTIFY= %USER user PACCT= 'pnode accounting data' PLEXCLASS= (pnode class, snode class) PNODE= primary node name PNODEID= (ID, pswd, newpswd) PRTY= n REQUEUE= Yes <u>No</u> RETAIN= Yes <u>No</u> Initial SACCT= 'snode accounting data' SNODE= secondary node name TCPNAME=tcpname;port SNODEID= (ID, pswd, newpswd) STARTT= ([date day] [,hh:mm:ssXM]) &symbolic 1= variable string 1 &symbolic 2= variable string 2 ...=... &symbolic N= variable string N

The following tables describes SUBMIT command parameters:

Parameter	Description
PROC = member	<p>Specifies the Process Library member name. This name is the member of the Public Process Library which contains the Process you are submitting. The member name is a 1-8 character alphanumeric string, with the first character alphabetic. This parameter is required if the Process resides in the Connect:Direct Public Process Library.</p> <p>The Connect:Direct Public Process Library PDS is specified by the DMPUBLIB ddname allocated to the TSO session or specified in the batch job stream.</p>
DSN = dsn dsn (member)	<p>Specifies the data set name or the name of the member of a PDS that contains the Process if the Process is not in the Connect:Direct Public Process Library. This parameter is required if the Process does not reside in the Connect:Direct Public Process Library.</p> <p>dsn is the data set name of a sequential file that contains the Process.</p> <p>dsn(member) specifies the PDS name and member name (in parentheses) that contains the Process. Specify either PROC or DSN; do not specify both.</p>
CASE = Yes <u>No</u>	<p>Specifies whether parameters associated with accounting data, user ID, password, and data set name in the command and in the Process are to be case sensitive. This parameter is optional.</p>
CLASS = n	<p>Determines the node-to-node session on which a Process can execute. See your Connect:Direct administrator for which class to specify. The range is 1-255. This parameter is optional.</p>

Parameter	Description		
DEBUG= trace	Specifies the 8-position trace setting for this Process. This parameter enables you to specify a settings for this trace for only this Process. This parameter is optional. Process The DEBUG Setting column in the following list shows trace values:		
	DEBUG Setting	Trace Type	
		Output DD	
	80000000	COPY Routine and RUN TASK trace	RADBDD01
	10000000	Full TPCB/SYMBOLICS from DMCBSUBM	DMCBSUBM
	08000000	Session manager trace	RADBDD05
	04000000	Separate trace per task (Example: "R0000005" to trace TASK 5)	Rnnnnnnn
	02000000	API session trace	RADBDD07
	01000000	DMGCSUB trace	RADBDD08
	00400000	TCQSH from DMCOPYRT	DMCOPYRT
	00200000	Make each SVC dump unique	N/A
	00100000	SECURITY trace control	SECURITY
	00040000	GETMAIN/FREEMAIN trace	RADBDD16
	00008000	I/O buffer trace	RADBDD21
	00004000	WTO all dynamic allocation parameters	RADBDD22
	00002000	Connect:Direct/Plex traces	
		ACTION queue manager trace	CDPLXACT
		CKPT queue manager trace	CDPLXCKP
		TCQ queue manager trace	CDPLXTCQ
		STATS queue manager trace	CDPLXSTA
		First REQUEST queue manager trace	CDPLXREQ
		Second and subsequent REQUEST queue manager trace. For example, "CDPLXR03" traces the third queue manager. The number of queue manager traces is based on the maximum number of servers from the asset protection (APKEY) file.	CDPLXRnn
		JOIN queue manager trace	CDPLXJOI
	00001000	Workload Balancing trace	CDPLXWLB
	00000100	In-storage tracing only	N/A
	00000080	RPL trace - long	RPLOUT
	00000040	RPL trace - short	RPLOUT
	00000020	Version 2 session trace	RADBDD33
	00000008	Logon exit trace	RADBDD35
	00000004	Logon processor trace	RADBDD36
	00000002	SCIP exit trace	RADBDD37
	00000001	SNMP Trap trace	SCTRAPDD

Parameter	Description
HOLD = <u>No</u> Yes Call	<p>Specifies whether or not the Processes are placed in the hold queue. This parameter is optional.</p> <p>No specifies that Connect:Direct does not place the Process in the hold queue, but places it in the WAIT for execution queue (EX). HOLD=NO is the default.</p> <p>Yes specifies that the Process remains in the hold queue until either a CHANGE PROCESS command releases the Process or a DELETE PROCESS command deletes the Process.</p> <p>When HOLD=YES and you specify a value for the STARTT parameter, Connect:Direct places the Process submitted in the hold queue.</p> <p>Note: When RETAIN=Y, Connect:Direct ignores the HOLD parameter.</p> <p>Call specifies that Connect:Direct is to place the Process in the hold queue until a session is established with the specified SNODE. This session is established by either another Process starting on the PNODE destined for the same SNODE or the SNODE contacting the PNODE. For example, a Process submitted with HOLD=NO establishes a session and causes execution of any Processes residing on the SNODE destined for this node that are submitted with HOLD=CALL.</p> <p>Note: To support dial-up connections from Connect:Direct/Windows platforms and release Processes serially, define the SNODE using a TCP/IP address of 0.0.0.0 (a null IP address). You may let the port number default since it will be resolved at connection time. Processes submitted to such nodes will default to HOLD=CALL status and can only be released by a NULL or ENABLE process from Connect:Direct/Windows. Releasing Processes to such nodes will immediately cause them to return to HOLD=CALL status without executing because the connection cannot be resolved. Processes initiated by such nodes will execute normally, but will not release the HOLD=CALL Processes. Checkpoint restart is supported for such nodes.</p>

Parameter	Description
MAXDELAY = [unlimited queued 0 hh:mm:ss]	<p>Indicates that the submit command waits until the submitted Process completes execution or the specified time interval expires. This parameter is optional. Do not use MAXDELAY for a submit within a Process—use only in SUBMIT commands.</p> <p>Note: If the Process does not complete within the time interval specified by queued or hh:mm:ss, the API returns SSPA006I, RC=4 and DMBATCH terminates with RC=48 (x'30').</p> <p>unlimited specifies that the submit command waits until the Process completes execution. This is the default when no parameters are specified.</p> <p>queued specifies that the submit command waits until the Process completes or 30 minutes, whichever occurs first.</p> <p>0 specifies that the submit command attempts to start a session for the submitted Process to execute on immediately. If Connect:Direct cannot establish a session, after all retries are exhausted, the Process is flushed and the submit command fails with the error <i>SVTM118I RC=52(x'34')</i>.</p> <p>Note: If Connect:Direct cannot establish a session after all retries are exhausted due to all available sessions on the remote node being in use, that is, when session attempts fail with error <i>SVTM080I SESSION (nnn) REJECTED pname (pnum) SNODE=remote.node</i>, the Process is flushed and the submit command fails with the error <i>SVTM118I RC=12(x'0C')</i>.</p> <p>Note: MAXDELAY=0 Processes will not use the intelligent retry feature. When a transfer to a remote node times out and retries, subsequent transfers to the same remote node will also time out and retry rather than being added to the wait queue.</p> <p>hh:mm:ss specifies that the submit command waits for an interval no longer than the specified hours, minutes, and seconds or until the Process completes, whichever occurs first.</p>
NEWNAME = newname	<p>Specifies the new name of the Process. The default is the label on the Process statement. This parameter is optional.</p>
NOTIFY = %USER user ID	<p>Specifies the user ID to receive Process completion messages. This parameter is optional.</p> <p>%USER specifies that the user who submitted the Process receives the completion messages, if the Connect:Direct user ID that the user is currently logged on with is the same as the TSO user ID. If the Connect:Direct user ID is different from the TSO user ID, the TSO user is not notified but the ID to which Connect:Direct is logged on will be notified.</p> <p>user ID specifies the TSO user ID that receives Process completion messages.</p> <p>Note: The NOTIFY capability is not supported across z/OS images in the sysplex environment. Connect:Direct cannot send Process completion messages across z/OS images in a sysplex.</p>
PACCT = 'pnode accounting data'	<p>Specifies the accounting data for the PNODE. The maximum length of the accounting data is 256 characters. If special characters are part of the accounting data, you must enclose the string in single quotation marks. This parameter is optional.</p> <p>This data overrides any accounting data specified on the SIGNON command and any accounting data specified in the Process statement of the submitted Process.</p>

Parameter	Description
PLEXCLASS = (pnode class, snode class)	<p>Specifies the class that directs the Process to only certain servers in a Connect:Direct/Plex. This parameter is only used in a Connect:Direct/Plex and is optional.</p> <p>You can designate each server in a Connect:Direct/Plex to support only certain PLEXCLASSES through the CDPLEX.PLEXCLASSES initialization parameter. You can then limit Processes to only those servers by specifying the PLEXCLASS in the Process definition.</p> <p>The pnode class controls which Connect:Direct Server runs the Process. The snode class controls what other node is used with the Process. The pnode class and snode class are each 1-8 characters long. An asterisk (*) indicates that the Process runs on any server with an asterisk designated in the CDPLEX.PLEXCLASSES initialization parameter. If you do not specify a PLEXCLASS, the network map is checked for a default PLEXCLASS. If the network map does not specify a default PLEXCLASS, then an asterisk is used as the default.</p> <p>If a Process must run on a specific Connect:Direct Server, specify the Connect:Direct Server name in this field. The Process only runs on that server.</p>
PNODE = primary node name	<p>Specifies the primary node of the Process. This parameter is optional.</p> <p>primary node name is a 1-16 alphanumeric character name defined in the network map. You can express the name in alphanumerics or nationals (@#)\$) with embedded periods.</p> <p>The node to which you submit the Process is always the PNODE. You do not need to specify this parameter. It defaults to the name of the node submitting the Process. PNODE is for documentation purposes only. For information on node name formats and values, see the SNODE parameter on page 84.</p>
PNODEID = (ID, pswd, newpswd)	<p>Specifies security user IDs and security passwords at the PNODE. This parameter is optional. ID specifies the security ID that Connect:Direct passes to a security exit for validations on the PNODE side of the Process. The range is 1-64 alphanumeric characters.</p> <p>pswd specifies the current security password. The security exit uses this parameter to validate the current security password on the PNODE side of the Process. It can contain 1-64 alphanumeric characters.</p> <p>newpswd specifies the new security password. The security exit uses this parameter to change the current security password to the new security password. The range is 1-64 alphanumeric characters.</p>
PRTY = n	<p>Specifies the Process priority in the Transmission Control Queue. High numbers indicate high priorities; low numbers indicate low priorities. Connect:Direct uses this priority only for Process selection; it does not affect the priority during transmission. The default is defined during your installation. The range is from 0-15. This parameter is optional.</p>

Parameter	Description
REQUEUE = Yes <u>No</u>	<p>Specifies whether Connect:Direct requeues a COPY step if an ABEND occurs during processing. This parameter is optional.</p> <p>Yes places the requeued Process in the HOLD queue with a status of HELD IN ERROR (HE). You can then take corrective action and restart the Process at the step that failed. Note that you must explicitly release the Process from the HOLD queue when the status is HELD IN ERROR (HE).</p> <p>No specifies the failing copy step is not requeued if it fails with an ABEND (such as X37). The remaining steps in the Process are allowed to execute. The default is NO. The value REQUEUE=No is forced in the case of a submit containing the MAXDELAY parameter.</p>
RETAIN = Yes <u>No</u> Initial	<p>Specifies whether or not Connect:Direct keeps a copy of the Process in the HOLD queue after the Process has executed. This parameter is optional.</p> <p>Yes keeps a copy of this Process in the HOLD queue after the Process executes. The copy of the Process does not execute until you release it through a CHANGE Process command. If you specify RETAIN=YES, Connect:Direct automatically holds the Process until you release it, unless you include the STARTT parameter in your Process. Use RETAIN in conjunction with STARTT to cause a Process to run repeatedly at a given time each day.</p> <p>No specifies that Connect:Direct is to delete the Process after execution. The default value for RETAIN is NO. The value RETAIN=No is forced in the case of a submit containing the MAXDELAY parameter.</p> <p>Initial specifies that Connect:Direct is to execute the Process every time Connect:Direct is initialized.</p> <p>Note: Do not code the STARTT parameter with the RETAIN=I parameter.</p>
SACCT = 'snode accounting data'	<p>Specifies the accounting data for the SNODE. The maximum length of the accounting data is 256 characters. If you include special characters with the accounting data, you must enclose the string in single quotation marks. This parameter is optional.</p> <p>This data overrides any accounting data specified on the SIGNON command and any accounting data specified in the Process statement.</p>
SNODE = secondary node SNODE=TCPNAME UDT33NAM = tcpvalue;port	<p>Specifies the secondary node used in this Process. This parameter is optional.</p> <p>secondary node name is a 1-16 alphanumeric name that is defined in the network map.</p> <p>The following characters are allowed: A-Z, 0-9, !, @, #, \$, %, &, {, }, +, -, ^</p> <p>Connect:Direct for z/OS does not accept the following characters for the SNODE: (,) =, \, ", ', <, >, , </p> <p>Use SNODE=TCPNAME UDT33NAM =tcpvalue to specify TCP/IP or UDT connections that are not defined in the Connect:Direct network map. tcpvalue can be a DNS name up to 255 characters or a 15-character IPv4 or 39-character IPv6 TCP/IP address.</p> <p>port is the TCP/IP port number. It can be up to 5 characters long.</p> <p>Note: If the TCPNAME keyword is used and the port is not specified, the TCP.IP.DEFAULT entry is used if the NODE is not defined in the Netmap.</p>

Parameter	Description
SNODEID = (ID, pswd, newpswd)	<p data-bbox="534 262 1235 289">Specifies security user IDs and security passwords at the SNODE.</p> <p data-bbox="534 300 1370 384">ID specifies the security ID that Connect:Direct passes to a security exit for validation on the SNODE side of the Process. It can contain 1-64 alphanumeric characters.</p> <p data-bbox="534 394 1419 625">pswd specifies the current security password. The security exit uses this parameter to validate the current security password on the SNODE side of the Process. It can contain 1-64 alphanumeric characters. In the case where the SNODE can Process a PassTicket password, the PNODE generates a PassTicket when only a SNODE user ID override is specified. The actual generation is contingent upon the information shared in the PNODE Authorization File and the option that generates the stage 2 security exit. The <i>Connect:Direct for z/OS Administration Guide</i> contains details for the Authorization File and stage 2 security exit characteristics.</p> <p data-bbox="534 636 1395 720">newpswd specifies the new security password. The security exit uses this parameter to change the current security password to the new security password. The range is 1-64 alphanumeric characters.</p>

Parameter	Description
STARTT = ([date day] [,hh:mm:ssXM])	<p>Specifies that Connect:Direct not execute the Process until a specified date or time. The date, day, and time are positional parameters. If you do not specify the date or day, place a comma before the time. This parameter is optional.</p> <p>Note: Do not code STARTT with RETAIN=INITIAL.</p> <p>date specifies the date to execute the Process. You can specify the day (dd), month (mm), and year (yy for 2-digit year and yyyy for 4-digit year). You can use periods or backslashes (/) to separate the components of a date value. You can omit the separators only for transfers between mainframe nodes. Use separators to guarantee transfers between all platforms.</p> <p>You can use the following date formats, according to which date order is specified in the DATEFORM initialization parameter:</p> <ul style="list-style-type: none"> ◆ DATEFORM=MDY specifies the date format as mm/dd/yy, mm/dd/yyyy, mm.dd.yy, or mm.dd.yyyy ◆ DATEFORM=DMY specifies the date format as dd/mm/yy, dd/mm/yyyy, dd.mm.yy, or dd.mm.yyyy ◆ DATEFORM=YMD specifies the date format as yy/mm/dd, yyyy/mm/dd, yy.mm.dd, or yyyy.mm.dd ◆ DATEFORM=YDM specifies the date format as yy/dd/mm, yyyy/dd/mm, yy.dd.mm, or yyyy.dd.mm <p>Valid Julian dates formats are yyddd, yyyyddd, yy/ddd, yyyy/ddd, yy.ddd, or yyyy.ddd</p> <p>If you specify a date without a time, the time defaults to 00:00.</p> <p>If RETAIN=YES, you cannot specify a date in the STARTT parameter.</p> <p>day specifies the day of the week that Connect:Direct is to release the Process for execution. Valid names are MOnday, TUEsday, WEdnesday, THursday, FRiday, SATurday, and SUNday. You can also specify TODAY, which releases the Process for execution today, or TOMORROW, which releases the Process for execution the next day.</p> <p>If you specify the day of the week with RETAIN=YES, the Process executes the same day every week.</p> <p>If you specify a day without a time, the time defaults to 00:00. A time of 00:00 means that if you submit a Process on Monday, with Monday as the only STARTT parameter, the Process does not run until the following Monday.</p> <p>hh:mm:ssXM indicates the time of day in hours (hh), minutes (mm), and seconds (ss) that Connect:Direct is to release the Process. You can set XM to AM or PM, or you can omit it if you are using a 24-hour clock. You need not specify minutes and seconds.</p> <p>You can express the time of day using the 24-hour clock or the 12-hour clock. If you do not use AM and PM, Connect:Direct assumes the 24-hour clock. If you use the 12-hour clock, you can express 01:00:00 hours as 1:00A, and 13:00 hours as 1PM. If you use the 24-hour clock, valid times are 00:00-24:00.</p> <p>You can also specify NOON, which releases the Process for execution at noon, or MIDNIGHT, which releases the Process for execution at midnight.</p> <p>If you code hh:mm:ssXM with RETAIN=YES, Connect:Direct schedules the Process for execution the same time every day.</p> <p>Note: When you specify both HOLD=YES and a STARTT value, the HOLD specification takes precedence. Connect:Direct places a Process submitted with HOLD=YES on the hold queue even if you specify a start time.</p>

Parameter	Description
&symbolic 1 = variable string 1	Specifies the default value substituted for the symbolic parameter in the Process. If you use a symbolic in the SUBMIT command, it overrides any default values you specify in the Process statement.
&symbolic 2 = variable string 2...	
&symbolic N = variable string N	Specify a null value by immediately following the equal sign with a comma. You must enclose a symbolic parameter containing special characters in single quotation marks. You can set an ampersand (&) symbolic parameter to another single ampersand symbolic parameter that is resolved during the first Process submission. Do not use identical symbolic names. The maximum variable string length is 256 characters.

Command Examples

Use these examples to help you gain a basic understanding of how you can use the SUBMIT command. For more advanced examples, see the [Connect:Direct Processes Web site](#).

Process Example

The following example shows the syntax for a Process named PAYROLL.

```
SUBMIT PROC=PAYROLL -
  HOLD=Y  RETAIN=Y -
  PACCT='1959, DEPT 27' -
  SACCT='1962, DEPT 24' -
  CASE=YES
```

In this example, the SUBMIT command was used to:

- ◆ Place the Process (named PAYROLL) in the HOLD queue.
- ◆ Retain a copy after it is released for execution.
- ◆ Create case sensitivity for fields associated with accounting data, user ID, password, and data set name in the SUBMIT command and in the PAYROLL Process.

Symbolic Substitution Example

In the following example, Connect:Direct resolves the symbolic &DSN in the Process COPYSEQ upon submission.

```
SUBMIT PROC=COPYSEQ -
  SNODE=CHICAGO, &DSN=MYFILE
```

Submitting Processes through the Batch Interface

To submit a Process through the batch interface, perform these steps:

1. Place your SUBMIT command in the DMBATCH job stream as described in Chapter 3, *The Batch Interface*.

2. Submit the job while Connect:Direct is running.

Note: If the Extended Submit Facility (ESF) is available, the DTF does not need to be running to execute your command.

3. Verify your results.

Submitting Processes through the IUI

The Interactive User Interface (IUI) enables you to create, edit, and submit Processes using the SB, DF, and CF options available on the PRIMARY OPTIONS MENU (see *The Primary Options Menu* on page 32).

IUI Submit Options

The SB, DF, and CF options available from the IUI enable you to perform the following submit options for Processes:

Option	Description
SB	Enables you to submit Processes from the Connect:Direct Public Process Library or a library you specify on the screen. You can also override certain parameters in the existing Process.
DF	Enables you to edit, create, and submit Processes residing in the Connect:Direct Public Process Library or a library you specify on the screen. You edit or create a Process using the ISPF editor.
CF	Enables you to dynamically build a COPY Process through fill-in-the-blank panels and submit it.

Validating Processes from the SB, DF, and CF IUI Options

From any of the IUI SUBMIT screens available through the SB, DF, and CF options, you can validate the Process syntax without submitting the Process, or you can submit the Process, which includes verifying the syntax, as explained in the following table:

Operation	Related Parameter or Command	Result
Validate Process syntax	<ul style="list-style-type: none"> ◆ Enable VALIDATE ONLY parameter ◆ Issue SUBV command 	<ul style="list-style-type: none"> ◆ The Process is displayed in the VIEW PROCESS screen. ◆ If the Process is valid, the message <i>Submit Process for validation command was successful</i> is displayed. ◆ If the Process is invalid, a message identifies the problems in the Process. You can press PF1 for information to troubleshoot the problem. ◆ You can press PF3 to return to the primary SB, DF, or CF screen.
Submit Process and validate syntax	<ul style="list-style-type: none"> ◆ Issue SUBMIT or SUB command 	<ul style="list-style-type: none"> ◆ If the Process is valid, it is executed and a Process number is displayed for reference. ◆ If the Process is invalid, a message describes the problem. You can press PF1 for information to troubleshoot the syntax errors.

Using the SB IUI Option to Submit a Predefined Process

The Interactive User Interface (IUI) enables you to submit predefined Processes using the SB option, which is available from the PRIMARY OPTIONS MENU (see *The Primary Options Menu* on page 32). You can submit a Process from the Connect:Direct Public Process Library or a library you specify and override certain parameters in the existing Process.

To submit a predefined Process from the Connect:Direct IUI:

1. Select option **SB** from the Primary Options Menu to display the SUBMIT PROCESS screen.

```

node.name          SUBMIT PROCESS          hh:mm
CMD==>            mm.dd.yyyy
                  yyyy.ddd

PROCESS NAME      ==>          (MEMBER Connect:Direct PROCESS LIBRARY)
OR
DATA SET NAME     ==>
AND MEMBER        ==>          (If dataset is PDS)
TRANSPORT         ==>          (DNS, LDNS, NET, TCP, or UDT33)
SECONDARY NODE    ==>
REQUEUE          ==>          (Y-YES, N-NO)
HOLD PROCESS      ==>          (Y-YES, N-NO, C-CALL)
RETAIN PROCESS    ==>          (Y-YES, N-NO, I-INIT)
PRIORITY          ==>          (0 TO 15)
NEW PROCESS NAME  ==>
CLASS             ==>          DEBUG      ==>
PLEXCLASS        ==>          (PNODE SNODE)
NOTIFY           ==> %USER
SECURITY OVERRIDE ==> N          VALIDATE ONLY ==> N
SCHEDULED        -- DATE --      -- TIME --
START            ==>              ==>
SYMBOLIC PARAMETERS ==>

DO YOU WANT VALUES FOR THIS PROCESS TO BE CASE SENSITIVE? ==> NO

```

2. Type the name of the Process to submit.
3. Take one of the following actions:
 - ◆ To submit the Process, press **Enter**.
 - ◆ To define additional parameters, continue with step 4.
4. Type any additional parameters of your SUBMIT command and press **Enter**.

For SUBMIT parameter descriptions not included in the following table, press **PF1** or see *The SUBMIT Command* on page 84.

Field	Description
TRANSPORT	<p>Specifies the method of transport to use for the file transfer.</p> <p>NETMAP—Causes Connect:Direct to search in the network map for the sending and receiving nodes. If you type a Transport of NETMAP, the SNODE keyword remains SNODE=secondary node.</p> <p>TCP—Indicates that the NODE NAME field contains an alias name or an IP address. Connect:Direct does not look in the network map for node information. If you type TCP in the Transport field, the SNODE keyword becomes SNODE=TCPNAME=tcvalue.</p> <p>UDT33—Indicates that the NODE NAME field contains an alias name or an IP address. Connect:Direct does not look in the network map for node information. If you type UDT33 in the Transport field, the SNODE keyword becomes SNODE=UDT33NAM=udtvalue.</p> <p>DNS—Enables you to enter a domain name up to 255 bytes.</p>
SECURITY OVERRIDE	<p>Enables you to override security information such as password and user ID for the primary and secondary node.</p> <p>Yes = Enable security override.</p> <p>No = Do not override security information.</p>
VALIDATE ONLY	<p>Enables you to validate the Process syntax without submitting the Process.</p> <p>Yes = Validate without submitting the Process.</p> <p>No = Do not validate without submitting the Process.</p>
CASE SENSITIVITY	<p>Specifies whether parameters associated with accounting data, user ID, password, and data set name in the Submit Process are to be case sensitive. The default is NO, which does not allow mixed case.</p>

- If you enable the SECURITY OVERRIDE parameter, type the security override information in the Connect:Direct SECURITY OVERRIDE screen.

```

node.name          Connect:Direct SECURITY OVERRIDE          TIME: hh:mm
CMD==>

PRIMARY NODE:

SECURITY ID: _____
PASSWORD      :
NEW PASS     :

SECONDARY NODE:

SECURITY ID: _____
PASSWORD      :
NEW PASS     :
```

- Press **Enter** to perform one of the following actions:
 - Submit the Process. See the table on page 95 for the possible results.

- ◆ View the results of the validate-only operation in the VIEW PROCESS panel, if you enabled the VALIDATE ONLY parameter. See the table on page 95 for the possible results.

Caution: Press **PF3** to exit the screen from which you submitted the Process. Each time you press **Enter**, the Process is submitted.

7. If you performed the validate-only operation, press **PF3** to return to the SUBMIT PROCESS screen.

Using the DF IUI Option to Create, Edit, and Submit Processes

The Interactive User Interface (IUI) enables you to modify, create, and submit Processes using the DF option, which is available on the PRIMARY OPTIONS MENU (see *The Primary Options Menu* on page 32). Using the DF option, you can:

- ◆ Modify a Process
- ◆ Create a new Process
- ◆ Submit a Process
- ◆ Copy statement models to create a Process

The Connect:Direct PUBLIC PROCESS LIBRARY contains Process statement models that you can use as templates for creating Connect:Direct Processes. The DF option enables you to copy Process statement models, edit them for your environment using the ISPF editor, and submit Processes residing in the Connect:Direct Public Process Library or in a library you specify.

The PUBLIC PROCESS LIBRARY contains commented files and uncommented files. File names of commented files are preceded by the at sign (@); file names with the pound sign (#) prefix do not include comments.

See *Validating Processes from the SB, DF, and CF IUI Options* on page 95 for details about how to validate the syntax of a Process with and without submitting it.

Modifying and Submitting a Process

To modify a Process using the ISPF editor and submit the Process to the DTF:

1. Choose option **DF** from the Primary Options Menu to display the Connect:Direct PROCESS DEFINITION screen.

```

node.name          Connect:Direct PROCESS DEFINITION SCREEN          hh:mm
CMD ==>

PROCESS MEMBER NAME ==> _____

PROCESS LIBRARY NAME ==> _____

PUBLIC PROCESS LIB ==> TEST1.PROCESS.LIB_____

DO YOU WANT VALUES FOR THIS PROCESS TO BE CASE SENSITIVE? ==> NO

ISPF EDITOR
MESSAGE:

```

2. Type the member name of the Process.
 3. Verify the library location and press **Enter**.
The PROCESS LIBRARY NAME field is required. If the member is located in the Connect:Direct PUBLIC PROCESS LIBRARY, type the library name that is displayed in the PUBLIC PROCESS LIB field in the PROCESS LIBRARY NAME field.
 4. After Connect:Direct displays the Process in ISPF edit mode, make the desired changes.
-
- Note:** For information on the valid content and syntax of Process statements, see the [Connect:Direct Processes Web site](#).
-
5. Press **PF3** to return to the PROCESS DEFINITION screen.
 6. Specify case sensitivity. Case sensitivity applies to the SUBMIT command and the Process.
 7. Take one of the following actions:
 - ◆ To validate the Process without submitting it, type **SUBV** and refer to the table on page 95 for information on the results.
 - ◆ To submit the Process, type **SUBMIT** (or **SUB**) on the command line and press **Enter**.

Creating and Submitting a New Process

To create and submit a new Process:

1. Choose option **DF** from the Primary Options Menu to display the Connect:Direct PROCESS DEFINITION screen.
2. Type the member name of the new Process.
3. Specify the name of the library where you want to save the new Process and press **Enter**.
The member location can be your private Process library (PROCESS LIBRARY NAME) or the Connect:Direct PUBLIC PROCESS LIBRARY.
4. After Connect:Direct displays an ISPF editor screen, type your new Process.

Note: For information on the valid content and syntax of Process statements, see the [Connect:Direct Processes Web site](#).

5. Press **PF3** to save the Process in the PROCESS LIBRARY you specified and return to the PROCESS DEFINITION SCREEN.
6. To validate the Process without submitting it, type **SUBV** on the command line and press **Enter**. See the table on page 95 for information about the results.
7. Specify case sensitivity. Case sensitivity applies to the SUBMIT command and the Process.
8. To submit the Process, type **SUBMIT** (or **SUB**) in the command line and press **Enter**.

Submitting a Process

To submit a Process to the DTF:

1. Choose option **DF** from the Primary Options Menu to display the Connect:Direct PROCESS DEFINITION screen.
2. Type **SUBMIT** (or **SUB**) on the command line.
3. Type the member name of the Process.
4. Verify that the library name that contains the member you want to submit is displayed in the PROCESS LIBRARY NAME field. If the member is located in the Connect:Direct PUBLIC PROCESS LIBRARY, type the library name that is displayed in the PUBLIC PROCESS LIB field in the PROCESS LIBRARY NAME field.
5. Specify case sensitivity. Case sensitivity applies to the SUBMIT command and the Process.
6. Press **Enter**. See the table page 95 for information about the results.

Using Statement Models to Create a Process

To copy a Process statement model to a new PDS member:

1. Choose option **DF** from the Primary Options Menu to view the Connect:Direct PROCESS DEFINITION screen.
2. Type the name of the new PDS member.

Note: The PUBLIC PROCESS LIB displays the partitioned data set (PDS) allocated by the signon CLIST or the TSO logon procedure.

3. Type the PROCESS LIBRARY NAME and press **Enter**.
4. Type **COPY** on the command line at the top of the blank member and press **Enter**.
5. When the ISPF Edit/View-Copy screen is displayed, type the member name of the PROCESS statement model you want to copy in the DATA SET NAME field and press **Enter**.

For example, to create your Process, first copy the PROCESS statement. Specify @PROCESS to use commented models from the PUBLIC PROCESS LIBRARY, or #PROCESS to use uncommented models in your Process.

6. To add a statement model, type **a** in the first column of the last line of the new member and press **Enter**.
7. Repeat step 4 through step 6 to copy additional statement models into the member following the PROCESS statement.
8. Edit the statement model or models for your environment according to the following guidelines:
 - ◆ Replace underscores with the appropriate parameter values.
 - ◆ Provide an appropriate Process name in the PROCESS statement.
 - ◆ Delete all lines that are not applicable.
 - ◆ Continuation marks are necessary on all but the last line of each statement model.
 - ◆ You can delete comment lines. They are optional.
9. Press **PF3** to save the changes and return to the Connect:Direct PROCESS DEFINITION screen.
10. Take one of the following actions:
 - ◆ To validate the Process without submitting it, type **SUBV** and press **Enter**. See the table on page 95 for information about the results.
 - ◆ To submit the Process, type **SUBMIT** on the command line and press **Enter**. See the table on page 95 for information about the results

Using the CF IUI Option to Generate a COPY Process

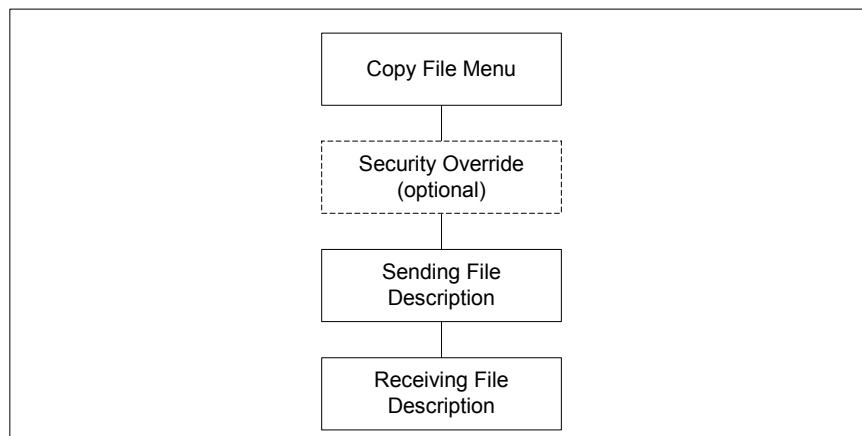
The CF option of the Interactive User Interface (IUI) enables you to dynamically generate a COPY Process through fill-in-the-blank panels and submit it.

The COPY FILE Menu of the CF option displays a series of screens that collect information to generate a SUBMIT PROCESS command. Connect:Direct submits the predefined Process, COPYCF, with the appropriate command parameters and variable substitutions based on your input. You can use four screens to create the COPYCF Process:

Screen Name	Description
COPY FILE Menu	Collects information for the SUBMIT COMMAND parameters, PNODE/SNODE information, data compression, and checkpoint information.
Security Override Menu	Collects information about user ID, password, and accounting data. This optional panel is displayed only if you request it on the first panel.

Screen Name	Description
SENDING FILE Menu	Collects information for the Process variables for the sending file.
RECEIVING FILE Menu	Collects the information for the Process variables for the receiving file.

The following diagram illustrates the relationship of these four screens.



See *Validating Processes from the SB, DF, and CF IUI Options* on page 95 for details about how to validate the syntax of a Process with and without submitting it.

To generate a COPY Process:

1. Select the **CF** option from the Primary Options Menu to display the COPY FILE Menu.

```

node.name          Connect:Direct COPY FILE MENU          TIME: hh:mm
CMD ==>

SENDING ENVIRONMENT: ZOS_____          TRANSPORT: NETMAP
NODE NAME: CD.NODE1_____
RECEIVING ENVIRONMENT: ZOS_____
NODE NAME: CD.NODE2_____
PROCESS NAME: COPYCF__
CLASS:            _____ (NUMERIC)          VALIDATE ONLY ==> N
HOLD:             N          (Y, N, OR C-CALL)
NOTIFY USER:    %USER_____
PRIORITY:        _____ (RANGE: 0 TO 15)
REQUEUE:         N          (Y OR N)
RETAIN ON TCQ:  N          (Y, N, OR I-INITIAL)
START DATE:      _____ TIME: _____
CHECKPOINT:      _____ (BYTE INTERVAL - NK|NM)
DEBUG:           _____ (HEXADECIMAL)
PLEXCLASS:       _____ (PNODE SNODE)
COMPRESS:        N_____ (Y, N, X-EXTENDED, X'XX', OR C'C')
EXT COMPRESSION LEVEL: _ WINDOW: _ MEMORY: _
OVERRIDE SECURITY: N          (Y OR N)

DO YOU WANT VALUES FOR THIS COPY TO BE CASE SENSITIVE? ==> NO___
  
```

2. If necessary, rename the COPYCF Process by typing a unique name in the PROCESS NAME field.
3. For descriptions of SUBMIT parameters not included in the following table, press **PF1** or see *The SUBMIT Command* on page 84.

Parameter Name	Description
SENDING ENVIRONMENT	Specifies the operating system or platform from which the file is transmitted. The values you place in the Sending Environment and Node Name fields determine which Sending File screen is displayed.
NODE NAME	Specifies the name of the Connect:Direct node from which the file is transmitted.
TRANSPORT	Specifies the method of TRANSPORT Connect:Direct uses for the file transfer. NETMAP—Causes Connect:Direct to look in the network map for the sending and receiving nodes. If you type a Transport of NETMAP, the SNODE keyword remains SNODE=secondary node. TCP—Indicates that the NODE NAME field contains an alias name or an IP address. Connect:Direct does not look in the network map for node information. UDT33—Indicates that the NODE NAME field contains an alias name or an IP address. Connect:Direct does not look in the network map for node information. If you type UDT33 in the Transport field, the SNODE keyword becomes SNODE=UDT33NAM=udtvalue. DNS—Enables you to enter a domain name up to 255 bytes Note: If you type TCP or DNS in the Transport field, the SNODE keyword becomes SNODE=TCPNAME=tcvalue.
RECEIVING ENVIRONMENT	Specifies the operating system or platform to which the file is transmitted. The values you place in the Receiving Environment and Node Name fields determine which Receiving File screen is displayed.
NODE NAME	Specifies the name of the node to which the file is transmitted.
PROCESS NAME	(optional) Specifies the name to be used under which the Process will be submitted. If you do not type a Process name, Connect:Direct provides the default name COPYCF.
VALIDATE ONLY	Enables you to validate the Process without submitting it. YES = Validate the Process, but do not submit it. NO = Do not validate before the Process is submitted.
CHECKPOINT	(optional) Specifies the byte interval for checkpoint support, which enables restart of interrupted transmissions at the last valid transmission point, avoiding the need to restart transmission from the beginning. K denotes thousands; M denotes millions. A checkpoint value of zero stops automatic checkpointing. For a complete description of this parameter, see the Connect:Direct Processes Web site .

Parameter Name	Description
COMPRESS	<p>(optional) Specifies whether and how the data being transmitted should be compressed.</p> <p>Y – Uses compression with X'40' (blank) as the PRIMEchar.</p> <p>N – Does not use compression.</p> <p>X – Uses Extended compression.</p> <p>X'xx' – Uses compression with the specified 'xx' hex vale.</p> <p>C'c' – Uses compression with the specified 'cc' character value.</p> <p>Note: Compression is CPU-intensive, and its effectiveness is data dependent. It should only be used if its benefits are known.</p>
EXT COMPRESSION LEVEL	<p>Note: The effects of changing the default values for the extended compression parameters (level, window, and memory) are not always predictable and can significantly increase CPU utilization. Before changing the default values, see the <i>Improving Performance</i> chapter in <i>Connect:Direct for z/OS Administration Guide</i> for more information.</p> <p>(optional) Specifies the value from 1–9 indicating the degree of compression to use. The default is 1, which usually provides sufficient compression.</p>
WINDOW	(optional) Specifies the value from 8–15 indicating the size of the compression buffer to use. The default of 13 equals 32 KB.
MEMORY	(optional) Specifies the value from 1–9 indicating the amount of memory used to maintain the compression state. The default of 4 equals 8 KB.
OVERRIDE SECURITY	<p>Enables you to change security information for the sending and receiving nodes.</p> <p>YES = Enable</p> <p>NO = Disable</p>
CASE SENSITIVITY	(optional) Specifies the case sensitivity not only to the SUBMIT command, but to the COPY Process itself. Connect:Direct provides for mixed case user input because some platforms allow it. The default is NO, which does not allow mixed case.

4. To override security information, type **Y** in the **OVERRIDE SECURITY** field.

```

node.name          Connect:Direct SECURITY OVERRIDE          TIME: hh:mm
CMD ==>

SENDING ENVIRONMENT: ZOS          TRANSPORT: TCP
NODE NAME:   CD.NODE1

SECURITY ID: _____
PASSWORD   :
NEW PASS   :
ACCOUNTING DATA: _____

RECEIVING ENVIRONMENT: ZOS
NODE NAME:   CD.NODE2

SECURITY ID: _____
PASSWORD   :
NEW PASS   :
ACCOUNTING DATA: _____

```

5. In the **SECURITY OVERRIDE** screen:
- Specify the security information for the node whose security you want to override and press **Enter**. This procedure assumes that you modify security information for both nodes.
 - Press **PF1** for a description of each field.

Note: For a complete description of the valid parameters of a COPY Statement and examples to help you fill in both the **SENDING FILE** and **RECEIVING FILE** screens, see the [Connect:Direct Processes Web site](#).

6. When the **SENDING FILE** screen for the environment and node name you specified is displayed, type the appropriate values and press **Enter**. Press **PF1** for a description of each field.

Note: If the data set name does not follow z/OS naming conventions, enclose the data set in single quotation marks.

An HFS file must begin with a slash (/) and can contain up to 251 bytes.

```

node.name          COPYFILE - SENDING FILE (z/OS or OS/390)          TIME: hh:mm
CMD ==>

NODE NAME: CD.NODE2
SENDING DSNAME: _____
_____
_____

UNIT PARAMETER: ( _____ )
LABEL PARAMETER: ( _____ )
VOLUME SERIAL(S): ( _____ )
RETAIN: _ (Y OR N)
DCB PARAMETER: ( _____ )
TYPE KEY: _____ MSVGP NAME: _____
SYSOPTS: _____
(PDS ONLY:)      REPLACE: Y (Y OR N)  ALIAS: Y (Y OR N)
SELECTION CRITERIA: ( _____ )
                  ( _____ )

EXCLUSION CRITERIA: ( _____ )
                  ( _____ )

```

7. When the RECEIVING FILE screen for the environment and node name you specified is displayed, type the appropriate values and press **Enter**. Press **F1** for a description of each field.

Note: If the data set name does not follow z/OS naming conventions, enclose the data set in single quotation marks.

An HFS file must begin with a slash (/) and can contain up to 251 bytes.

```

node.name          COPYFILE - RECEIVING FILE (z/OS or OS/390)          TIME: hh:mm
CMD ==>

NODE NAME: Q1A.ZOS.V4600
RECEIVING DSNAME: _____
_____
_____

DISPOSITION: ( NEW , CATLG_ , _____ )
UNIT PARAMETER: ( _____ )
VOLUME SERIAL(S): ( _____ )
RETAIN: _ (Y OR N)  COUNT: ___ (1-255)
DCB PARAMETER: ( _____ )
LABEL PARAMETER: ( _____ )
SPACE: ( _____ )
TYPE KEY: _____ MSVGP NAME: _____
SYSOPTS: _____
SMS: DATA CLASS: _____ STORAGE CLASS: _____ MGMT CLASS: _____
AVERAGE RECORDS: _ DSNTYPE : _____
VSAM ORGANIZATION: ___ KEY LENGTH: ___ KEY OFFSET: _____
LIKE DSNAME: _____
SECURITY MODEL: _____
GENERIC MODEL: ___ (YES OR NO)

```

8. Press **Enter**.

- ◆ If you enabled the Validate Only parameter, a VIEW PROCESS screen with your Process is displayed. See the table on page 95 for information on the results.
- ◆ If you did not select the Validate Only option, the Process is executed. See the table on page 95 for information on the results.

Caution: Press **PF3** to back out of each screen until the Connect:Direct Primary Options Menu is displayed. Each time you press **Enter**, the Process is submitted.

Controlling Processes in the TCQ

This chapter describes how to control and monitor Processes after they are submitted to the TCQ, and includes the following topics:

- ◆ Controlling Processes with Commands
- ◆ Modifying a Process in the TCQ with CHANGE PROCESS
- ◆ Suspending, Flushing, and Deleting Processes
- ◆ Examining Processes in the TCQ

Note: You can also use the Connect:Direct Browser User Interface to perform some of the procedures in this chapter. To learn more about the Connect:Direct Browser User Interface, see the user's guide on the Connect:Direct Browser User Interface CD-ROM or available online from the Sterling Commerce Documentation Library.

See the [Connect:Direct Processes Web site](#) for instructions on building a Process and Chapter 7, *Building, Modifying, and Submitting Processes* of this book for instructions on submitting the Process to the TCQ.

Controlling Processes with Commands

Connect:Direct Processes consist of statements with parameters that provide instructions for copying files, running jobs and programs, and altering the sequence of Process step execution. Use the Process Control commands described in this chapter to manipulate these Processes while they are in the TCQ.

Setting Selection Criteria

The Process control commands have common parameters and IUI screen layouts. All Process control commands allow you to select Processes by Process name, number, and submitter. (Some allow additional criteria.) The WHERE parameter groups the selection arguments, enabling you to make common changes or selections. The command applies to all Processes which match the criteria. For example, if you indicate a Process name PROCA with no further qualification, all Processes named PROCA are affected by the command.

When you specify two or more WHERE subparameters, you further qualify the selection so that all conditions must be satisfied. In this way, you can adequately qualify your selection if you have non-unique Process names or numbers. For example, if you specify in your selection criteria a Process name of PROCA and Process number of 16, the Process number of PROCA must be 16 for the command to apply.

If you specify multiple Processes by using the list option on the subparameter, the position of the arguments in the list is significant. Elements in each list must correspond by position. For example, PNAME=(PROCA, PROCB), PNUM=(16,17) requires that PROCA be Process number 16 and PROCB be Process number 17 for the command to apply.

On the screens, you can type a list of Process names, numbers, and submitters (user ID and node ID) to obtain the same results that the WHERE parameter list provides in batch. Connect:Direct builds the appropriate command from the screen. Connect:Direct accesses the corresponding Process names and numbers and submitters (user ID and node) from left to right to construct the list form of this command.

Modifying a Process in the TCQ with CHANGE PROCESS

Use the CHANGE PROCESS command to modify the parameters of a Process when the Process is in the TCQ in a nonexecuting state. Release a held Process or restart a failed Process with this command. (You can also release a held Process through the Operator Table and Selected Process screens by typing an R next to the Process Name.)

Command Format

The CHANGE PROCESS command has the following format and associated parameters. Required parameters and keywords are in bold print.

Label	Command	Parameters
(optional)	CHange PROCESS	WHERE (NODE= node name (list) PNAME= name (list) PNUMber= number (list) SUBmitter= (node name, user ID) (list)) CASE= Yes No CLASS= n DEBUG= trace bits DEST= destination node HOLD= Yes No Call NETMAP.REFRESH PLEXCLASS=(pnode class, snode class) PRTY= n RELEASE RESTART= [NO FIRST = volume sequence number FIRST = SER = volume serial number LAST = volume sequence number LAST = SER = volume serial number VOLCNT = n] RETAIN= Yes No Initial STARTT= ([date day] [,hh:mm:ssXM])

The following table describes the parameters for the CHANGE PROCESS command:

Parameter	Description
WHERE (NODE = node name (list) PNAME = name (list) PNUMber = number (list) SUBmitter = (node name, user ID) (list)	Specifies which Process(es) to change. Name multiple Processes in the command using the selection criteria method described on page 110 if you have to make similar changes to many Processes. This parameter is required.
NODE = node name (list)	Specifies the snode or a list of snodes where the Processes to be changed are running. If you specify a list of snodes, enclose them in parentheses separate each snode with a comma. You can use this subparameter with the HOLD parameter to put all Processes on a node on hold if the node is unavailable, or if problems exist with the node. You can restart all Processes later using the RESTART parameter with the NODE= subparameter.
PNAME = name (list)	Specifies the name of the Process to be changed or a list of Process names enclosed in parentheses and separated by commas. This parameter is optional.

Parameter	Description
PNUMBER = number (list)	Specifies the number of the Process to be changed or a list of Process numbers enclosed in parentheses and separated by commas. The range is 1-99999.
SUBmitter = (node name, user ID) (list)	Specifies the node name and user ID of the user that submitted the Process to be changed. Specify a list of SUBmitter IDs by enclosing the IDs in parentheses and separating them by commas.
CASE = Yes No	Specifies whether parameters associated with accounting data, user ID, password, and data set name are case sensitive. The designation refers only to the command, not to the Process itself. See page 22 for a general overview of case sensitivity. This parameter is optional.
CLASS = n	Determines the node-to-node session on which a Process executes. Consult your Connect:Direct administrator for instructions concerning which class to specify. The range is 1-255. This parameter is optional.
DEBUG= trace settings for this Process	Specifies the 8-position trace setting for this Process. This parameter enables you to specify a trace for only this Process. The table on page 86 lists acceptable trace values. This parameter is optional.
DEST = destination node	Specifies a new destination node. This parameter changes the node that this Process communicates with. This parameter is optional.
HOLD = Yes No Call	<p>Specifies whether or not the Process is placed in the hold queue. This parameter is optional.</p> <p>Yes specifies that the Process remains in the hold queue until one of the following events occurs:</p> <ul style="list-style-type: none"> ◆ CHANGE PROCESS releases the Process ◆ DELETE PROCESS deletes the Process <p>When you specify both HOLD=YES and a STARTT value, the HOLD specification takes precedence. Therefore, Connect:Direct places a Process submitted with HOLD=YES on the hold queue even if you specified a start time.</p> <p>No specifies that Connect:Direct does not place the Process in the hold queue, but places it in the WAIT for execution queue (EX). HOLD=NO is the default.</p> <p>Call specifies that Connect:Direct is to place the Process in the hold queue until a session is established with the specified SNODE. This session could be established by either another Process starting on the PNODE destined for the same SNODE or the SNODE contacting the PNODE. For example, a Process submitted HOLD=NO establishes a session and causes execution of any Processes residing on the SNODE destined for this node that are submitted with HOLD=CALL.</p> <p>Note: Connect:Direct ignores the HOLD parameter if RETAIN=Y.</p>
NETMAP.REFRESH	<p>Refreshes the network map for Processes waiting execution in the Wait queue, after an UPDATE NETMAP command is issued for this node. (See the <i>Maintaining the Network Map</i> chapter of the <i>Connect:Direct for z/OS Administration Guide</i> for a description of the UPDATE NETMAP command.) This parameter is optional.</p> <p>Do not use the NETMAP.REFRESH parameter if the network map protocol is changed (for example, changing a node from LU0 to LU6.2). If you change the protocol for a particular node, you must delete and resubmit all waiting Processes for that node.</p>

Parameter	Description
PLEXCLASS = (pnode class, snode class)	<p>Specifies the class that directs the Process to only certain servers in a Connect:Direct/Plex. This parameter does not apply to a Connect:Direct Stand-alone Server. This parameter is optional.</p> <p>You can designate each server in a Connect:Direct/Plex to support only certain PLEXCLASSES through the CDPLEX.PLEXCLASSES initialization parameter. You can then limit Processes to only those servers by specifying the PLEXCLASS in the Process definition.</p> <p>The pnode class controls which Connect:Direct Server runs the Process. The snode class controls what other node is used with the Process.</p> <p>The pnode class and snode class are each 1-8 characters long. An asterisk (*) indicates that the Process runs on any server with an asterisk designated in the CDPLEX.PLEXCLASSES initialization parameter. If you do not specify a PLEXCLASS, the network map is checked for a default PLEXCLASS. If the network map does not specify a default PLEXCLASS, then an asterisk is used as the default.</p> <p>If a Process must run on a specific Connect:Direct Server, specify the Connect:Direct Server name in this field. The Process only runs on that server.</p>
PRTY = n	<p>Specifies the Process priority in the Transmission Control Queue. High numbers indicate high priorities; low numbers indicate low priorities. Connect:Direct uses this priority only for Process selection; it does not affect the priority during transmission. The default is set by the PRTYDEF global initialization parameter. The range is from 0-15. This parameter is valid for LU0 only. This parameter is optional.</p>
RELEASE	<p>Releases the Process for execution from the queue where it is currently residing. Either specify RELEASE or omit it. Releases or RELEASE=no is not valid. This parameter is optional.</p>
RESTART = [NO FIRST = volume sequence number FIRST=SER=volume serial number LAST = volume sequence number LAST = SER = volume serial number VOLCNT = n]	<p>Specifies the conditions for restarting an interrupted Process.</p> <p>Use the CHANGE PROCESS command to cause Connect:Direct to restart a data transmission at the last checkpoint position taken before the interruption or at a previous checkpoint position. Connect:Direct restarts the copy step at the position you specify. This parameter is optional.</p> <p>NO specifies that the copy step restarts at the beginning of the transmission.</p> <p>FIRST=volume sequence number specifies that the copy step restarts at the beginning of the volume designated by the volume sequence number.</p> <p>FIRST=SER=volume serial number specifies that the copy step restarts at the beginning of the volume serial given.</p> <p>LAST=volume sequence number specifies that the copy step restarts at the end of the volume designated by the volume sequence number. LAST specifies that the copy step restarts at the last block on the volume if the output is disk or the last checkpoint on the volume if the output is tape.</p> <p>LAST=SER=volume serial number specifies that the copy step restarts at the end of the volume serial given. LAST specifies that the copy step restarts at the last block on the volume if the output is disk or the last checkpoint on the volume if the output is tape.</p> <p>VOLCNT=n specifies that the volume count on the interrupted copy step is changed to the value specified. You can use this parameter to increase the number of output volumes if the copy step is interrupted because the volume count of that step is too small.</p>

Parameter	Description
RETAIN = Yes No Initial	<p>Specifies whether or not Connect:Direct keeps a copy of the Process in the HOLD queue after the Process has executed. This parameter is optional.</p> <p>Yes keeps a copy of this Process in the HOLD queue after the Process executes. The copy of the Process does not execute until you release it through a CHANGE Process command. If RETAIN=YES is specified, Connect:Direct automatically holds the Process until you release it, unless you include the STARTT parameter in your Process. Use RETAIN in conjunction with STARTT to cause a Process to run repeatedly at a given interval.</p> <p>No specifies that Connect:Direct is to delete the Process after execution. The default value for RETAIN is NO.</p> <p>Initial specifies that Connect:Direct is to execute the Process every time Connect:Direct is initialized. Do not code the STARTT parameter with the RETAIN=I parameter.</p>
STARTT = ([date day] [,hh:mm:ssXM])	<p>Specifies that the Process not execute until a specified date or time. See page 92 for a complete explanation of the STARTT parameter. This parameter is optional. When changing the date, day, or hh:mm:ssXM, you must respecify even the values that do not change. You cannot specify TODAY or TOMORROW for the day subparameter.</p>

Note: Processes with a QUEUE or STATUS of PR cannot be changed with the CHANGE PROCESS command.

Command Examples

The following CHANGE PROCESS command places the Processes named PAYROLL and BILLING in the hold queue.

```
CH PROC WHERE ( -
  PNAME=(PAYROLL, BILLING)) -
  HOLD=Y
```

In the following example, the Process named PAYROLL is changed so that the new destination node is DALLAS1 and the Process executes every Friday.

```
CH PROC WHERE (PNAME=PAYROLL) -
  DEST=DALLAS1 -
  RETAIN=Y -
  STARTT=(FR)
```

In the following example, the Process called PAYROLL with the Process number of 60584 has the RETAIN status changed so that the Process is deleted after execution.

```
CH PROC WHERE (PNUM=60584, PNAME=PAYROLL) -
  RELEASE -
  RETAIN=N
```

Issuing CHANGE PROCESS through the Batch Interface

To use the CHANGE PROCESS command from the Batch Interface:

1. Place your commands in the DMBATCH job stream as described in Chapter 3, *The Batch Interface*.
2. Submit the job while Connect:Direct is running.

A report similar to the following is displayed.

```

=====
                           CHANGE PROCESS
=====
PROCNAME PROCNUM_ SUBMITTER_NODE__ SNODE.NAME_____ MESSAGE_____
USERID_____
-----
DMNOTEST 271      node.name          node.name          DEST CHANGED
USER01
-----

```

3. Check this report to verify your results.

Issuing CHANGE PROCESS through the IUI

Use the CHANGE PROCESS screen to change a Process and its associated parameters. To issue the CHANGE PROCESS command from the Connect:Direct IUI, perform the following steps:

1. Select option CP from the Primary Options Menu to display the CHANGE PROCESS screen.

```

node.name          CHANGE PROCESS          hh:mm
CMD ==>

PROCESS NUMBERS: ==> _____ ==> _____ ==> _____ ==> _____
PROCESS NAMES:   ==> _____ ==> _____ ==> _____ ==> _____
NODES            ==> _____ ==> _____ ==> _____
USER IDS:        USER NODES:
==> _____
==> _____
==> _____
==> _____

PLEXCLASS        ==> _____ (PNODE SNODE)
DESTINATION NODE ==> _____ DEBUG ==> _____
PRIORITY         ==> _____ (0 TO 15)
CLASS            ==> _____
HOLD PROCESS     ==> _ ( 'Y'-YES, 'N'-NO, 'C'-CALL)
RELEASE PROCESS  ==> _ ( 'Y'-YES, 'N'-NO )
RETAIN PROCESS   ==> _ ( 'Y'-YES, 'N'-NO, 'I'-INIT)
REFRESH FROM NETMAP ==> _ ( 'Y'-YES, 'N'-NO )
SCHED. START DATE ==> _____ TIME ==> _____
RESTART ALTERATIONS ==> _____
DO YOU WANT THE VALUES OF THIS REQUEST TO BE CASE SENSITIVE? ==> NO

```

Connect:Direct selects the corresponding Process names and numbers and submitters (user ID and node) from left to right to construct the list form of this command. This list fully qualifies your search criteria.

2. Type at least one of the parameters or a combination of parameters as described in the CHANGE PROCESS command syntax section on page 110 or press the **PF1** key for Help. Connect:Direct changes the parameters for the Processes that match the search criteria. The case sensitivity designation refers only to the command parameters on the screen, not to the parameters of the Process itself.

Suspending, Flushing, and Deleting Processes

The SUSPEND PROCESS, FLUSH PROCESS, and DELETE PROCESS commands have similar formats and parameters. The following table describes each command:

Process	Description
SUSPEND PROCESS	Terminates execution and puts an executing Process in the Hold queue. Connect:Direct places the Process in HOLD status. You can restart the Process through the RELEASE parameter of the CHANGE PROCESS command, or by using the R option on the Operator Table screen or the Selected Process screen.
FLUSH PROCESS	Terminates an executing Process. Note that if you specify RETAIN=YES, the flushed Process remains in the TCQ for execution at the next scheduled time. If you specify RETAIN=NO, Connect:Direct removes the Process from the TCQ, and you must resubmit it if you want it to execute.
DELETE PROCESS	Enables you to specify the name of a nonexecuting Process and remove it from the TCQ. Note: FLUSH PROCESS stops an executing Process.

If you do not specify FORCE for the FLUSH or SUSPEND command, an indicator notifies the program executing on behalf of the Process that a FLUSH or SUSPEND command was issued for the Process. If that program is not in control (for example, if it is waiting on a request outside of Connect:Direct code to complete), then it does not see the FLUSH or SUSPEND indicator and the Process is not flushed or suspended; otherwise, the program looks for the FLUSH or SUSPEND indicator and takes the appropriate action.

When you specify FORCE, the action taken depends on the STATE and SUBSTATE of the Process for which you issued the FORCE FLUSH or FORCE SUSPEND. Determine the STATE and SUBSTATE of the Process by doing a SELECT PROCESS or SELECT TASK command.

Note: For a Process in a VTAM I/O STATE and a SUBSTATE of SEND or RECEIVE, the FLUSH or SUSPEND command is implemented whether you specify FORCE or not. That is not true for the other states.

SUSPEND, FLUSH, or DELETE PROCESS Command Format

The SUSPEND PROCESS, FLUSH PROCESS, and DELETE PROCESS commands have the following format and associated parameters. Required parameters and keywords are in bold print.

Label	Command	Parameters
(optional)	FLUSH PROCess DELETE PROCess SUSPEND PROCess	WHERE (PNAME= name (list) PNUMber= number (list) SUBmitter= (node name, user ID) (list) FORCE) Note: The FORCE parameter is not valid for DELETE PROCESS.
		CASE = Yes No

The following table describes the parameters:

Parameter	Description
WHERE (PNAME = name (list) PNUMber = number (list) SUBmitter = (nodeid, user ID) (list) FORCE)	Specifies which Process to suspend, delete, or flush. Name multiple Processes in the command using the search criteria method described on page 110. WHERE is the only required parameter for the SUSPEND PROCESS, DELETE PROCESS, and FLUSH PROCESS commands. Its subparameters are optional. However you must specify at least one of the PNAME, PNUMBER, and SUBMITTER subparameters. PNAME = name (list) specifies the name of the Process or a list of Process names enclosed in parentheses and separated by commas. PNUMBER = number (list) specifies the number of the Process selected or a list of Process numbers enclosed in parentheses and separated by commas. The range is 1-99999. SUBmitter = (nodeid, user ID) (list) specifies the nodeid and user ID of the user that submitted the Process. Specify a list of SUBmitter IDs by enclosing the IDs in parentheses and separating them by commas. FORCE enables you to suspend or flush a Process that is waiting for unavailable resources. If the initial suspend or flush attempt fails, retry the command with FORCE. Note: FORCE parameter is not valid for DELETE PROCESS. If FORCE is specified for a Process executing on an LU6.2 session, it can terminate the session immediately, and Connect:Direct does not exchange the Process statistics between nodes.
CASE = Yes No	Specifies whether parameters associated with accounting data, user ID, password, and data set name are case sensitive. The designation refers only to the command, not the Process itself. This parameter is optional.

Note: Processes with a QUEUE or STATUS of PR cannot be flushed or suspended with the FLUSH or SUSPEND PROCESS command.

Command Examples

This section contains examples of the SUSPEND, DELETE, and FLUSH commands.

SUSPEND Process

The following command suspends any executing Process with a submitter node ID of DALLAS and a submitter user ID of SMITH.

```
SUSPEND PROC WHERE (SUB=(DALLAS, SMITH))
```

The following command suspends the Process named PAYROLL, which is Process number 514 and the Process named COPY100, which is Process number 575.

```
SUS PROCWHERE ( -
PNAME=(PAYROLL,COPY100), -
PNUM=(514,575))
```

DELETE Process

The following command deletes Processes with the number 60584.

```
DEL PROC WHERE (PNUM=60584)
```

The following command deletes all Processes submitted by CHUCK at the node DALLAS.MVS.

```
DEL PROC WHERE (SUB=(DALLAS.MVS, CHUCK))
```

The following command deletes all Processes in the PR queue. When specifying QUEUE= PR, no other selection criteria are allowed.

```
DELETE PROCESS WHERE (QUEUE=PR)
```

FLUSH Process

The following command flushes executing Processes submitted by SMITH at the node DALLAS.

```
FLUSH PROC WHERE (SUB=(DALLAS, SMITH))
```

The following command flushes the Process named PAYROLL which has Process number of 514 and the Process named COPY100 which has the Process number of 575.

```
FLUSH PROC WHERE ( -
PNAME=(PAYROLL,COPY100), -
PNUM=(514,575))
```

Issuing the Commands through the Batch Interface

To use the `SUSPEND PROCESS`, `DELETE PROCESS`, or `FLUSH PROCESS` commands from the Batch Interface, perform the following steps:

1. Place commands in the `DMBATCH` job stream as described in Chapter 3, *The Batch Interface*.
2. Submit the job while `Connect:Direct` is running.
3. Verify the results.

Issuing the Commands through the IUI

To issue commands through the IUI:

1. Access the function that you want as described in the following table.

Function	Access Method
Suspend an executing Process	<ul style="list-style-type: none"> ◆ Select option PS from the Primary Options Menu. ◆ Type SUS at the Primary Options Menu command prompt ◆ Type a P next to the Process Name on the Operator Table, the Operator Table - Executing Queue, or the Selected Process screens.
Delete a nonexecuting Process	<ul style="list-style-type: none"> ◆ Select option DP from the Primary Options Menu. ◆ Type a D next to the Process Name on the Operator Table or the Selected Process screens.
Flush a Process	<ul style="list-style-type: none"> ◆ Select option FP from the Primary Options Menu. ◆ Type an F next to the Process Name on the Selected Process screen.

The following screen shows an example of the Suspend an Executing Process screen. The Delete A Nonexecuting Process screen and the Flush Process screen are similar except that the `FORCE` option is not on the Delete A Nonexecuting Process screen.

```

node.name                SUSPEND AN EXECUTING PROCESS                hh:mm
CMD ==>

PROCESS NUMBERS:
==> _____ ==> _____ ==> _____ ==> _____
==> _____ ==> _____ ==> _____ ==> _____

PROCESS NAMES:
==> _____ ==> _____ ==> _____ ==> _____
==> _____ ==> _____ ==> _____ ==> _____

FORCE: ('Y'-YES, 'N'-NO) FORCE SUSPENDING A PROCESS ON AN LU 6.2 SESSION
==> _                MAY TERMINATE THE SESSION IMMEDIATELY WITHOUT
                        STATISTICS OF THE PROCESS BEING EXCHANGED

USER IDS:                USER NODES:
==> _____
                        ==> _____
==> _____
                        ==> _____
==> _____
                        ==> _____

DO YOU WANT THE VALUES OF THIS REQUEST TO BE CASE SENSITIVE? ==> NO

```

Connect:Direct builds the SUSPEND PROCESS command from this screen, and selects the corresponding Process names and numbers and submitters (user ID and node) from left to right to construct the list form of this command. This list fully qualifies your search criteria.

2. Type at least one of the parameters or a combination of parameters as described in the SUSPEND PROCESS command syntax section on page 116, or press the **PF1** key for Help. Connect:Direct suspends the Processes which match your search criteria.

Note: The case sensitivity designation refers only to the command parameters on the screen, not to the parameters of the Process itself.

3. To minimize the risk of deleting, suspending, or flushing a Process accidentally, you may be required to confirm a Process request before it is executed. If you are required to confirm delete, flush, and suspend commands, a panel similar to the following screen is displayed. Do one of the following:
 - ◆ To confirm the operation, type Y on the command line. Connect:Direct performs the function and returns to the previous screen.

- ◆ To return to the previous screen without performing the function, press **PF3**. To change the Process numbers, select this option before executing the command.

```

DMDELCN2                                     DATE: YYYY/MM/DD
Connect:Direct Confirm Delete Command        TIME: HH:MM
CMD ==>

You have requested the following process(es) be deleted, are you sure this is
what you want to do?

Reply Yes, No or CANCEL or Press PF3 to Cancel the Request

Process Numbers  ==> 123          ==>          ==>
                ==>          ==>          ==>

Process Names    ==>          ==>          ==>
                ==>          ==>          ==>

USER IDS:  USER NODES:
  ==>
                ==>
  ==>
                ==>
  ==>
                ==>
  ==
                ==>

  __ Do not display this Confirm Delete prompt again.

```

Note: If you see the option, Do not display this Confirm Delete prompt again, you can type an X next to this option to turn off the Confirm prompt so that it will not display again during the current session.

Examining Processes in the TCQ

Use **SELECT PROCESS** to look over Processes in the TCQ. You can specify the search criteria and the form in which the information is presented (file, printout, table, or screen display).

Note: Use the **SELECT STATISTICS** command to determine the outcome of a completed Process.

SELECT PROCESS Command Format

The **SELECT PROCESS** command has the following format and parameters. The required parameters and keywords are in bold print. Default values are underlined>.

Note: The syntax and parameters for the **VIEW PROCESS** command are identical

Label	Command	Parameters
(optional)	SElect PROCess or View PROCess	WHERE (DEST= node (list) PNAME= name (list) PNUMber= number (list) QUEUE= All queue name SERVER= server name STATUS= Process status (list) SUBmitter= (node name, user ID) (list)) CASE= Yes No FILE PPrint <u>TABLE</u>

The following table describes the parameters.

Parameter	Description
WHERE (DEST = node (list) PNAME = name (list) PNUMber = number (list) QUEUE = All Exec Hold Timer Wait STATUS = Process status (list) SERVER = server name (list) SUBmitter = (node name, user ID) (list))	Specifies which Processes to select. Name multiple Processes in the command using the search criteria method described on page 110. The subparameters PNAME, PNUMber, and SUBmitter are optional, but you must specify at least one. WHERE is the only required parameter for the SELECT PROCESS command. Not all its subparameters are required. DEST = node (list) specifies the destination node name of the Process you are selecting or a list of destinations enclosed in parentheses and separated by commas. PNAME = name (list) specifies the name of the Process or a list of Process names enclosed in parentheses and separating by commas.

Parameter	Description
	<p>PNUMBER = number (list) specifies the number of the Process you are selecting or a list of Process numbers enclosed in parentheses and separated by commas. The range is 1-99999.</p> <p>QUEUE = All queue name specifies Process selection based on the TCQ. Values are: All specifies selection of a Process from all queues. This value is the default. queue name can be one of the following:</p> <p>E Executing queue H Hold queue PR Retain queue T Timer queue W Wait queue</p> <p>SERVER = server name (list) selects Processes on the specified Connect:Direct Servers. The server name is a 1-8 character name assigned to each server in a Connect:Direct/Plex through the CDPLEX.SERVER initialization parameter.</p> <p>STATUS = Process status (list) specifies Process selection by status value or a list of status values in parentheses separated by commas. Values include:</p> <p>H=All Held Processes HC=Held for Call HE=Held for Error HI=Held Initially HO=Held by Operator HP=Held due to Process error HR=Held Retain HS=Held for Suspension PR=Retained after Execution R=All Restart Processes RA=Held for Restart due to Allocation error RH=Restart Held (may be due to the Intelligent Retry feature) W All Waiting Processes (including Retry) WC=Wait for Connection WT=Waiting for Transport WX=Waiting for Connect:Direct Server</p> <p>SUBmitter = (node name, user ID) (list) specifies the nodeid and user ID of the user that submitted the Process. Specify a list of SUBmitter IDs by enclosing the IDs in parentheses and separating them by commas.</p>
CASE = Yes No	Specifies whether parameters associated with accounting data, user ID, password, and data set name are case sensitive. The designation refers only to the command, not the Process itself. See page 22 for a general overview of case sensitivity. This parameter is optional.
FILE PPrint <u>TABLE</u>	Specifies the form in which the information is presented. TABLE is the default. See page 24 for detailed information about the output format produced by each of the these parameters. This parameter is optional.

SELECT PROCESS Command Example

The following command searches for all Processes submitted by SMITH at the node CD.DALLAS.

```
SEL PROC WHERE (SUB=(CD.DALLAS, SMITH))
```

Issuing SELECT PROCESS through the Batch Interface

To use the SELECT PROCESS command from the Batch Interface, perform the following steps:

1. Place commands in the DMBATCH job stream as described in Chapter 3, *The Batch Interface*.
2. Submit the job while Connect:Direct is running.
3. Verify your results.

Issuing SELECT PROCESS through the IUI

Use the Select Process screen to specify the Processes that you want to display and how you want them displayed. To issue the SELECT PROCESS command in the Connect:Direct IUI, perform the following steps:

1. Select option SP from the Primary Options Menu to display the Select Process screen.

```

node.name          SELECT PROCESS
CMD ==> O                                               hh:mm

CMD:  O ... OPERATOR TABLE  S ... OPERATOR TABLE/EXEC QUEUE STATUS
      P ... PRINT REPORT     D ... DISPLAY REPORT      V ... VIEW PROCESS

QUEUE:           ==> _ (A-ALL, W-WAIT, E-EXECUTE, H-HOLD, T-TIMER, P-PR)

PROCESS NUMBERS: ==> _____ ==> _____ ==> _____ ==> _____
PROCESS NAMES:   ==> _____ ==> _____ ==> _____ ==> _____
SERVER NAMES:    ==> _____ ==> _____ ==> _____ ==> _____

STATUS:          (HO, HR, HI, HE, HC, HP, HS, PR, RH, RA, WC, WX, WT, H, R, W)
                ==> _      ==> _      ==> _      ==> _

DESTINATION NODES:
                ==> _____ ==> _____
                ==> _____ ==> _____

USER ID:         NODE ID:
                ==> _____
                ==> _____
                ==> _____

DO YOU WANT VALUES FOR THIS REQUEST TO BE CASE SENSITIVE? ==> NO

```

2. To select the Processes, specify the parameters to use as search criteria. These parameters are described in *SELECT PROCESS Command Format* on page 121.

Note: The case sensitivity designation refers only to the command parameters on the screen, not to the parameters of the Process itself.

3. To select the destination for your output, type the letter representing one of the following output options on the command line:
 - ◆ **O** accesses the Operator Table and creates a one line summary of each selected PNODE Process. For more information, see *Viewing and Controlling a Process through the Operator Table* on page 125.
 - ◆ **S** builds an Operator Table consisting only of Processes that are currently executing. The table shows how much data is transmitted for a COPY step. For more information, see *Accessing Execution Queue Status on the Operator Table* on page 127.
 - ◆ **V** displays the content of the matching Processes within the TCQ. For more information, see *Viewing the Content of Processes Matching Your Search Criteria* on page 129.
 - ◆ **P** prints a report of the selected Processes.
 - ◆ **D** displays a report, similar to the following figure.

```

BROWSE -- temporary file name          ----- LINE 00000000 COL 001 080
COMMAND ==>                          SCROLL ==> PAGE
***** TOP OF DATA *****
=====
                          SELECT PROCESS
=====
Prc Name   => PDEXCLUD           Queue    => EXEC
Prc Num    => 4                   Status   => EX
Prc Debug  => 00003021
Subnode    => SC.MVS.USERID       Time     =>
Other Node => SC.MVS.USERID       Date     =>
Server     => SERVER1            Day      =>
Userid     => USERID
Prtty     => 10      Retain => N   Ret proc => NONE
Class     => 2                State   => FILE I/O
Step Name  => EXCLUDE           Xmit St =>
FUNCTION   => COPY              MEMBER  => V2SPEC
Sent: Blks => 371 Recs => 0 RU's => 1,075

-----
Prc Name   => PDEXCLUD           Queue    => EXEC
Prc Num    => 4                   Status   => PR.CNTL

```

Viewing and Controlling a Process through the Operator Table

The Operator Table displays the PNODE Processes which satisfied your selection criteria; it does not display information about Processes submitted from another node.

1. After you type **O** on the command line of the Select Process screen, the Operator Table/Executing Queue screen is displayed.

The first two characters of the QUEUE indicate the queue, the second two characters indicate the status value. The OTHER.NODE is the name of the nonsubmitting node in the session.

```

node.name
----- OPERATOR TABLE -----
==> Q                               SCROLL ==> PAGE
OPTION  PNAME      PNUMBER  SUBMITTER.NODE----- OTHER.NODE ----- QUEUE
S+  SERVER  USERID
-----
-          DELPROC    100      CD.DALLAS      CD.LOSANGELES  TI HR
  SERVER1  USER01
-          HLDPROC    101      CD.DALLAS      CD.NEWYORK      TI WC
  SERVER1  USER02
-          RELPROC    102      CD.DALLAS      CD.LOSANGELES  HO HI
  SERVER1  USER01
-          SELPROC    105      CD.DALLAS      CD.BOSTON       HO HR
  SERVER1  USER01
-          SUSPROC    104      CD.DALLAS      CD.NEWYORK      EX EX
  SERVER1  USER03

```

2. Type an option in the OPTION column next to the Process name (PNAME) and press **Enter**. The following table describes each option.

Option	Description
H	Place a nonexecuting Process on the Hold queue.
D	Delete a nonexecuting Process from the queue. For more information, see <i>Issuing the Commands through the IUI</i> on page 119.
R	Release a held Process.
P	Suspend a Process from the executing queue. For more information, see <i>Issuing the Commands through the IUI</i> on page 119.
S	Select a Process for detailed display. For more information, see <i>Viewing and Controlling a Process through the Selected Process Screen</i> on page 128.
V	View a Process in the executing queue. For more information, see <i>Viewing the Content of Processes Matching Your Search Criteria</i> on page 129.

Note: When you release a Process which contains RETAIN=YES, it is copied and assigned a new Process number.

3. If you are required to confirm a delete or suspend Process request before it is executed, respond to the Confirm Command prompt.

Note: If message ID SOPA006I is displayed on the Operator Table screen next to the Process name under the OPTION column, it indicates that the Process can no longer be found. The Process was probably in execute status when another function was selected, and the screen was not refreshed. Screens are not automatically refreshed. Press **Enter** to refresh the screen.

- To view any message text, type **END** to leave the Operator Table and return to the Select Process screen. Then type **M** on the command line, and press **Enter**.

Accessing Execution Queue Status on the Operator Table

- After you type **S** at the command line of the Select Process screen, the Operator Table/Executing Queue displays Processes that are currently executing. The Operator Table/Executing Queue screen shows how much data is transmitted by the COPY step, including:
 - Blocks for block-mode transmissions
 - Records for record-mode transmissions
 - RUs (request/response units)
 - I/O bytes
 - VTAM bytes
 - The compression factor

The following figure is a sample of the screen and shows that two Processes are executing.

```

node.name                                     Row 1 to 2 of 2
-----OPERATOR TABLE/EXECUTING QUEUE-----
==> Q                                         SCROLL ==> PAGE
OPTION      PNAME          PNUMBER      SUBMITTER.NODE-- OTHER.NODE----- QUEUE
  SERVER      USERID
-----
      S3      BENCHRC          272          Q1A.ZOS.V4600    Q1A.ZOS.V4600    ~P EX
      USER01
      Blks => 6              Recs => 0              RUs => 173
      I/O bytes => 177,138          Compression
      VTAM bytes => 177,152          Factor => 0.0%

      S3      BENCHRC          272          Q1A.ZOS.V4600    Q1A.ZOS.V4600    EX EX
      USER01
      Blks => 7              Recs => 0              RUs => 174
      I/O bytes => 195,440          Compression
      VTAM bytes => 178,176          Factor => 8.8%

***** Bottom of data *****

```

- Type an option in the OPTION column next to the Process name (PNAME) and press **Enter**. The following table describes each option.

Option	Description
P	Suspend a Process in the executing queue.

Option	Description
S	Select a Process for detailed display. For more information, see <i>Viewing and Controlling a Process through the Selected Process Screen</i> on page 128.

Viewing and Controlling a Process through the Selected Process Screen

1. From the Operator Table, type **S** in the OPTION column next to the name of the Process to display the Selected Process screen. This screen displays details about the Process that you selected from the Operator Table.

The following figure is a sample of the screen which displays all available information about the particular Process you selected from the Operator Table.

```

node.name                SELECTED PROCESS                hh:mm
CMD ==>

  Select process command was successful.
Process Name => COPYCF      Number => 1          Step => COPYSTEP
Other Node   => SC.IRV.TEST1  DEBUG => 00003001  Status =>EX
Commid      => M3QRC763
Function     => COPY        Sub State => SEND   State =>VTAM I/O
Server      => SVTSERV1     PLEXCLASS => (
Submitter   => SC.IRV.TEST1                SENDING SIDE
Userid      => TEST1
Scheduled Time =>          Date =>          Day =>
Queueing Prty => 10       Class => 3          Retain => NO
Submitted Class=> NONE    Max Class => NONE   Sess.ID=> PNODE
Session restrt => 0      Dyn rstrt => 0      RouteID=>
Last Msgid  =>          Last RC => 00000000    RetProc=>
Secure=> UN-AUTHENTICATED  Signature => N Encryption=> NONE
Sending File => CDTEST1.TESTFILE.THRICE.V10
Receiving File => TEST1.TESTFL.THRICE
Volume seq no. => 1      Volser => USER01   TRN => 00030200
Blks        => 8        Recs => 2,486     RUs => 198
I/O bytes   => 223,360
VTAM bytes  => 202,752                Compression Factor => 9.2%

```

2. Type any of the following Process control commands at the command line, and press **Enter**. The following table describes each option:

Command	Description
P	Suspend the currently executing Process
D	Delete a nonexecuting Process
H	Place a nonexecuting Process in the HOLD queue
R	Release a nonexecuting Process
Enter	Monitor the currently executing Process execution

Command	Description
C	Change Other Node, Queueing Priority, Scheduled Time, Scheduled Day, and Retain Status in a nonexecuting Process

- To change parameters, move the cursor to the appropriate field on the screen, make the changes that you want, and press **Enter**.
- If you are required to confirm a Process request, respond to the Confirm Delete/Suspend Command prompt.
- If the Process completes execution while the Selected Process screen is displayed, press **Enter** to return to the Operator Table screen.

Viewing the Content of Processes Matching Your Search Criteria

- To view output on the VIEW PROCESS screen, do one of the following:
 - Type a V on the command line on the Select Process screen after specifying the parameters to use as search criteria. These parameters are described in *SELECT PROCESS Command Format* on page 121. Also, for more information, see *Issuing SELECT PROCESS through the IUI* on page 124.
 - Type a V in the OPTION column next to the Process name displayed on the Operator Table/Executing Queue screen. (The PNODE Processes which satisfied your selection criteria on the Select Process screen are displayed on the Operator Table/Executing Queue screen.) See *Viewing and Controlling a Process through the Selected Process Screen* on page 128.

The following shows the sample output of the VIEW PROCESS command:

```

=====
                        VIEW PROCESS
=====
                PROCESS NAME: TOEND      PROCESS NUMBER: 1
=====
TOEND      PROCESS  SNODE=CD.NODE          -
                PNODE=CD.NODE             -
                HOLD=YES                   -
                RETAIN=NO                  -
                CLASS=1                    -
                PRTY=2                     -

STEP01     COPY FROM                       -
                (SNODE                     -
                DSN= 'USER01.JCL.LIB'      -
                DISP= (SHR)                -
                )                           -
                TO                          -
                (PNODE                     -
                DSN= 'USER01.JCL.LIBX6'    -

```

Process Queuing and Recovery

This chapter provides information about Process queues and recovery, including the following topics:

- ◆ The Transmission Control Queue (TCQ)
- ◆ Process Recovery

The Transmission Control Queue (TCQ)

Connect:Direct stores submitted Processes in the TCQ. The TCQ controls Process execution. It consists of two inter-dependent VSAM Relative Record Data Sets (RRDS), and an in-memory queue which controls access. The two data sets are the TCQ and the TCX, which is a space map for the TCQ.

Logical Queues

The TCQ has the following logical queues:

- ◆ Wait
- ◆ Execution
- ◆ Hold
- ◆ Timer
- ◆ Process Retention

Access the queues and manipulate Processes using the following Connect:Direct commands:

- ◆ CHANGE PROCESS
- ◆ DELETE PROCESS
- ◆ FLUSH PROCESS
- ◆ SELECT PROCESS
- ◆ SUSPEND PROCESS

Refer to Chapter 8, *Controlling Processes in the TCQ*, for more information the on each of these Connect:Direct commands.

Queuing Parameters

The RETAIN, HOLD, and STARTT parameters queue Processes as described in following table.

Parameters	Queue	Comments
None	Wait	Process remains on the Wait queue until Connect:Direct can start a session with the SNODE. At this time, the Process moves to the Execution queue.
RETAIN=INITIAL	Hold	Process automatically executes each time Connect:Direct is initialized with TCQ=WARM. This setting is useful for Processes that contact other Connect:Direct nodes each time Connect:Direct completes initialization. This action causes any work queued on the remote node for the local node to begin.
RETAIN=YES	Hold	A copy of the Process is kept in the Hold queue after it has executed. The Process does not execute again until it is released by a CHANGE PROCESS command. You can use RETAIN=YES, combined with STARTT, to run a Process at a periodic interval. For example, RETAIN=YES and STARTT=(Tuesday, 3pm) starts the Process every Tuesday at 3 pm; RETAIN=YES and STARTT=(, 12:00) starts the Process each day at noon.
HOLD=YES	Hold	Process remains in the Hold queue until someone releases the Process.
HOLD=CALL	Hold	Process is automatically moved from the Hold queue to the Wait queue when the SNODE contacts the node on which the Process resides.
STARTT	Timer	When the scheduled time and date arrive, the Process is put on the wait queue and is available for execution. You can use RETAIN=YES, combined with STARTT, to run a Process at a periodic interval. For example, RETAIN=YES and STARTT=(Tuesday, 3pm) starts the Process every Tuesday at 3 pm; RETAIN=YES and STARTT=(, 12:00) starts the Process each day at noon. You can also use the Timer queue for session retry and file allocation retry based on Connect:Direct initialization parameters specified by a particular installation. When you exhaust retry limits, the Process is moved to the Hold queue with an HE status.

When you submit a Process, Connect:Direct puts the Process in the appropriate logical queue based on Process statement routing parameters listed in the preceding table. See *Process Execution Example* on page 79 for a step-by-step illustration of how Connect:Direct executes a Process.

TCQ Status and State Values

The SELECT PROCESS command displays Connect:Direct status values, Connect:Direct task state values, and transport state values. These are defined in the following table:

Value	Definition
Connect:Direct Status Value	Each Process on the TCQ has an associated Connect:Direct status value. This status value has a unique meaning determined by which queue the Process is in. The SELECT PROCESS command displays Connect:Direct status values.
Connect:Direct Task State Values	When a Process is in the Execution queue, the SELECT PROCESS command also displays a Connect:Direct task state value. The state values are provided for information purposes. You cannot modify or control them using Connect:Direct commands. The task state value shows the current state of the Process. Usually, Connect:Direct tasks are waiting for completion of a service such as File I/O, Connect:Direct locked resource, or VTAM I/O.
Transport State Values	<p>When a Process is on the Execution queue, the SELECT PROCESS command can also display a transport state value, depending on timing. The state values are provided for informational purposes. You cannot modify or control them using Connect:Direct commands. For example, if the Process is currently in a VTAM I/O state, the SELECT PROCESS output can show a further state value such as send or receive.</p> <p>While a Process is executing, the SELECT PROCESS command displays the number of file blocks or records and VTAM request/response units (RUs) sent or received. This display gives you an indication of the status of COPY statements on the Process.</p>

The following sections describe the Connect:Direct status values, task state values, transport state values, and commands for each queue.

Wait Queue

The following table shows the Connect:Direct status value for the Wait Queue.

Status Value	Explanation
WC	Waiting for Connection. This status is the initial queue status when a Process is submitted without HOLD or RETAIN specified. This status means the Process is ready to execute as soon as possible. Process is ready to run but other Processes are executing with the same SNODE, and no other sessions are available. This Process runs as soon as an eligible session is available. If you find a Process in this state, it is most likely for this reason. Processes are also in this state when Connect:Direct is quiesced.
WT	Waiting for transport. This status is the status of a Process when the transport protocol is not available. This Process runs as soon as the transport protocol is available.
WX	Waiting for server. This status is the status of a Process waiting for a Connect:Direct Server to become available. This Process runs as soon as an eligible Connect:Direct Server is available. An eligible Connect:Direct Server is an active server that supports the Process PLEXCLASS and the transport protocol (SNA, TCP, UDT, or CTCA). The transport protocol must also be available on the server for it to be eligible.

The following table shows the applicable commands for the Wait Queue.

Command	Description
Change Process	Modifies Process attributes
Delete Process	Removes the Process from the queue
Select Process	Displays Process status and state

Execution Queue

The following table shows the Connect:Direct status values for the Execution Queue.

Status Value	Explanation
EX	Node is in Process control and executing the displayed Process.
WC (Waiting for Connection)	Process control is in negotiation while the two nodes determine which Process executes next, based on priority.
PR.CNTL	This node is not in Process control. This status occurs: (1) during Process negotiation where highest priority on either node runs next, or (2) when the current node is the SNODE during Process execution.
SS	Session with other node is being started.

The following table shows Connect:Direct task state values for Processes in the Execution Queue.

Task State	Explanation
DISPATCH	Task is waiting to dispatch
INACTIVE	Task can be dispatched but inactive
VTAM I/O	Task is waiting on VTAM request
P=SNODE	PNODE equals SNODE task
NETEX I/O	Waiting on NETEX I/O request
VSAM I/O	Waiting on VSAM I/O request
MISC	Miscellaneous I/O, such as a WTO
FILE I/O	Non-VSAM I/O
LOCK	Waiting for Connect:Direct locked resource
SUBTASK	Waiting on a subtask, such as open or close, allocation, security, or a RUN TASK program
RUNNING	Executing instructions

Task State	Explanation
TCA SCAN	TCA scan
TIMER	Waiting for timer event
ATTACH	Waiting for attached tape drive
ALLOCATE	Waiting for allocation to complete
MOUNT	Waiting for tape mount
OPEN	Waiting for OPEN to complete

The following table shows the Subtask state values for the Execution Queue. The first characters of the subtask request state indicate the session protocol such as TCP or LU6.2.

Subtask State	Explanation
ADOPT V2	Performing Adopt
CALL V2	Performing Call
INIT V2	Performing Initialization
CLEANUP	Performing Cleanup
HANGUP	Performing Hangup
SEND V2	Performing Send
SEND RSP	Performing Send Response
SEND SIG	Performing Send Signal
RECV V2	Performing Receive
ANSWER V2	Performing Answer
IOCTL V2	Performing I/O Control
GET BUF	Performing I/O Control Get Buffer
RDY RCV	Performing I/O Control Ready Receive

The following table shows the transport state values for the Execution Queue.

Transport State	Explanation
NO SESSION	No VTAM session
SESSION EST	Session with another node being established
NO REQUEST	No VTAM request outstanding
RECEIVE	Waiting on VTAM RECEIVE request
OPEN	Waiting on VTAM OPEN request
CLOSE	Waiting on VTAM CLOSE request
SETLOGON	Waiting on VTAM SETLOGON request
REQSESS	Waiting on a request session request
OPNDST	Waiting on an open destination request
CLSDST	Waiting on a close destination request
GENCB EXTLST	Waiting on a GENCB EXLIST
GENCB ACB	Waiting on a GENCB ACB
GENCB NIB	Waiting on a GENCB NIB
GENCB RPL	Waiting on a GENCB RPL
REJSESS	Waiting on a reject session request
SESSIONC	Waiting on a session cancel request
INQUIRE	Waiting on an inquire request
OPNSEC	Waiting on an open secondary request
RSHUTD	Waiting on a request shutdown request
SIMLOGON	Waiting on a simulate logon request
SND RESPONSE	Waiting on a send response request
WAIT FOR +DR	Waiting on a definite response request
VTAM I/O	Waiting on VTAM request
SEND	Waiting on VTAM send request
DACTSESS	waiting on LU6.2 deactivate session request
RCVFMH5	Waiting on LU6.2 receive FMH-5 request
REJECT	Waiting on LU6.2 reject conversation request
ACTSESS	Waiting on LU6.2 activate session request
ALLOC ALLOCD	Waiting on LU6.2 allocate conversation until available request

Transport State	Explanation
PRERECV	Waiting on LU6.2 prepare to receive request Note: In a Connect:Direct/Plex environment, the LU6.2 connection protocol does not enable the Connect:Direct Manager to redirect work to one of its servers. The remote node must address the server on which you want to run an LU6.2 Process. To do this, specify the node name and VTAM address of the Connect:Direct Server on which the Process is to run in the remote server's network map. Use the same CDPLEX.SERVER.NODE and CDPLEX.VTAM specified for the local initialization parameters for the Connect:Direct Server you are trying to address.
CNOS	Waiting on LU6.2 change number of session request
DELLOC	Waiting on LU6.2 deallocate conversation request
SEND CONFRMD	Waiting on LU6.2 send confirmation request
6.2 RECEIVE	Waiting on LU6.2 receive request
6.2 SEND	Waiting on LU6.2 send request
SEND ERROR	Waiting on LU6.2 send error request
DISPLAY	Waiting on LU6.2 display session limit request
DEFINE	Waiting on LU6.2 define session limits request
ALLOC IMMED	Waiting on LU6.2 allocate conversation immediately request

The following table shows the allocate state values for the Execution Queue.

Allocate State Value	Explanation
SVC99	Performing SVC99
CAT SEARCH1	CAMLST locate
READ VTOC	CAMLST search

The following shows the applicable command descriptions for the Execution Queue.

Applicable Command	Description
Flush Process	Terminates and deletes an executing Process
Select Process	Displays Process status and state
Suspend Process	Terminates and moves an executing Process on the hold queue

Hold Queue

The following table shows the Connect:Direct Status values for the Hold Queue.

Status Value	Explanation
HC (Held for Call)	<p>Process is submitted with HOLD=CALL specified.</p> <p>A session started from either node causes Connect:Direct to place this Process on the wait queue in WC status, and eventually in the execution queue (EX) when the first Process finishes.</p> <p>Note: If the SNODE is defined with a null IP address, the Processes will be released serially. Refer to the HOLD parameter of the SUBMIT command on page 87 for further information.</p>
HE (Held in Error)	The Process was submitted and received an error unrelated to allocation or session errors. The Process is being checkpointed and REQUEUE=YES is specified. A common error that causes Connect:Direct to place the Process in HE status is an out of space condition (Sx37 ABEND).
HI (Held Initially)	Process is submitted with HOLD=YES specified.
HO (Held by Operator)	An exception response is sent from the other node during FMH exchanges at Process negotiation or step termination. It also occurs if an FMH is invalid or is sent out of sync, or if the remote node is not defined in the network map.
HP (Held due to Process Error)	An exception response is sent from the other node during EXIT exchanges at Process negotiation or step termination. HP also occurs if an EXIT is invalid, sent out of sync, or if the remote node is not defined in the Network map.
HR (Held Retain)	Process is submitted with RETAIN=YES specified or being held for Process retry (intelligent retry).
HS (Held for Suspension)	The operator issued a SUSPEND PROCess command. You can release the Process later. The Process will run (Held for Suspension) when Connect:Direct is recycled.
RA (Held for Restart Due to Allocation Error)	During Process execution, an allocation error occurred that matched those specified in the initialization parameters. This status enables the Process to restart after the allocation problem is resolved.
RH (Restart Held)	Either the Process is being held awaiting a connection to an inactive node by the Intelligent Session Retry feature (see <i>Intelligent Session Retry</i> on page 78) or a checkpointed Process is executing when an error such as a lost session or an I/O error occurred. This status enables the copy to restart when the session is reestablished.
WC (Wait For Connection)	Session establishment is attempted, including retries if specified, and failed. The current Process is put on the wait queue (and later EX Q) if a session with that node is established later. It also can be released.

The following table shows the applicable commands for the Hold Queue.

Applicable Command	Description
Change Process	Modifies Process attributes
Delete Process	Removes the Process from the queue
Select Process	Displays Process status and state

Process Retention Queue

The following table shows the Connect:Direct Status value for the Process Retention Queue.

Status Value	Explanation
PR (Process Retention)	A completed Process from the PNODE is placed in the Process retention queue, if you enable the ability to retain the Process after it has executed.

The following table shows the applicable commands for the Process Retention Queue.

Applicable Command	Description
Delete Process	Removes the Process from the queue
Select Process	Displays Process status and state

Timer Queue

The following table shows the Connect:Direct Status values for the Timer Queue.

Status Value	Explanation
RE (Retry)	The session with the SNODE is in the retry state. The number of and interval between retries is specified in the initialization parameters. The Process can be in retry status for session establishment or for an allocation error.
WC (Wait For Connection)	The Process is submitted with a start time or date (STARTT) that has not expired. When the STARTT is reached, the Process is put on the wait queue for scheduling to the EX Q.

The following table shows the applicable commands for the Timer Queue.

Applicable Command	Description
Change Process	Modifies Process attributes
Delete Process	Removes the Process from the queue

Applicable Command	Description
Select Process	Displays Process status and state

Process Recovery

Connect:Direct provides facilities to recover from most errors that occur during Process execution. Recovery from the point of failure is usually accomplished quickly. The following types of errors can occur during normal operation:

- ◆ Link failure terminates a session between Connect:Direct systems
- ◆ File I/O error occurs during Process execution
- ◆ Connect:Direct abends because of a hardware or other error
- ◆ TCQ Corruption

Connect:Direct provides the following facilities to address errors:

Facility	Description
Session establishment retry	<p>When one or more Processes run with a node, Connect:Direct establishes a session with that node and begins execution. If Connect:Direct cannot start the session, Connect:Direct retries the session establishment. The initialization parameters, MAXRETRIES and WTRERIES, determine the number of retries and the interval between retries.</p> <p>If Connect:Direct cannot establish a session after all retries are exhausted, the Process is placed in the Hold queue in the TCQ with a status of Waiting for Connection (WC). When a session is established with the other node, all other Processes are scanned and the highest priority Process is executed after the previous Process is finished.</p>
VTAM automatic session retry	<p>If Process execution is interrupted because of a VTAM session failure, Connect:Direct automatically attempts to restart the session. This recovery facility uses the same parameter values as the session establishment retry facility.</p> <p>If Connect:Direct cannot establish the session, the Process that is executing and any other Processes that are ready to run with the other node are placed in the Hold queue with a status of Waiting for Connection (WC).</p>
TCQ/TCX Repair Utility	<p>When the TCQ becomes corrupt because of an outage or other circumstance, Connect:Direct may abend in production or during the next DTF initialization. The Connect:Direct administrator can use the TCQ/TCX repair utility to remove ambiguous or corrupt data and avoid having to cold start the DTF and reinitialize the TCQ, thus losing any Processes left in the TCQ.</p> <p>See the <i>Connect:Direct for z/OS Administration Guide</i> for information on how to use this batch utility.</p>
Process step checkpoint	<p>As a Process executes, Connect:Direct records which step is executing in the TCQ. If Process execution is interrupted for any reason, the Process is held in the TCQ. When the Process is available for execution again, Connect:Direct automatically begins execution at that step.</p>

Facility	Description
COPY statement checkpoint/restart	<p>For physical sequential files and partitioned data sets, Connect:Direct collects positioning checkpoint information at specified intervals as a COPY statement executes. Checkpoints are taken for each member that is transferred within a PDS, regardless of the checkpoint interval. If the copying procedure is interrupted for any reason, you can restart it at the last checkpoint position.</p> <p>Note: Whenever a Process step is interrupted and restarted, some data will be retransmitted. Statistics records for the Process step will reflect the actual bytes transferred, and not the size of the file.</p> <p>The COPY statement checkpoint/restart works in conjunction with step restart. The restart is automatic if Connect:Direct can reestablish a session based on the initialization parameter values for MAXRETRIES and WTRERIES. See <i>COPY Statement Checkpoint/Restart Facility</i> on page 141 for more information.</p> <p>The CHANGE PROCESS command can also invoke the checkpoint/restart facility. See Chapter 8, <i>Controlling Processes in the TCQ</i>, for instructions on how to use the CHANGE PROCESS command.</p> <p>Note: Checkpoint/restart is not supported for I/O exits at this time.</p>

COPY Statement Checkpoint/Restart Facility

The checkpoint/restart facility includes the following elements:

- ◆ Initialization parameters
- ◆ Checkpoint/restart file
- ◆ Copy statement checkpoint parameters

Initialization Parameters

Connect:Direct uses the following initialization parameters with the COPY statement checkpoint/restart facility.

Parameter	Description
CKPT.MODE	<p>Specifies whether the checkpoint function is performed when Connect:Direct is transferring a file in record mode or block mode. (Record mode transfer is used when reblocking of the output file is specified.) It also specifies whether the checkpoint function is performed for partitioned data sets and if so, what type. Automatic checkpointing of VSAM files is supported.</p> <p>This parameter does not apply to TCP/IP and LU6.2 connections. With TCP/IP and LU6.2 connections, checkpointing is based on the checkpoint interval requested (regardless of the data set type or whether it is in block or record format) and on both nodes' ability to perform the checkpoint function. When both nodes agree to do checkpointing, the checkpoint interval is controlled by the sending node while checkpoint record filing is done on the receiving node.</p>

Parameter	Description
CKPT	<p>Specifies the default interval for checkpointing when it is not specified on the COPY statement.</p> <p>You should specify the CKPT value as a multiple of the value specified for the V2.BUFSIZE initialization parameter. If you do not, performance can seriously deteriorate.</p> <p>Note: Be aware of additional overhead associated with specifying too small a checkpoint interval, particularly when transferring large files.</p> <p>Note: For sequential files, try to avoid using a CKPT value less than: $BLKSIZE * NCP * 10$ where NCP is the number of buffers for reading data from or writing data to a sequential data set using BSAM (see <i>Improving Performance</i> in the <i>Connect:Direct for z/OS Administration Guide</i> for more information).</p>
CKPT.DAYS	<p>Specifies the amount of time that checkpoint records are kept if they are not deleted. Connect:Direct automatically deletes checkpoint records when a Process is restarted and runs to a successful completion.</p>
CKPTDSN	<p>Specifies the name of the checkpoint/restart file that holds checkpoint records during execution of the COPY statement.</p>

See the *Global Initialization Parameters* appendix in the *Connect:Direct for z/OS Administration Guide* for detailed information about these parameters.

Checkpoint/Restart File

The Connect:Direct checkpoint/restart file contains positioning information for both files involved in executing a COPY statement. Connect:Direct maintains the checkpoint records throughout data transmission and deletes them when a transmission completes successfully.

Note: Checkpoints take place on the receiving end of a transfer. During restart, this information is exchanged with the sender so that appropriate positioning can take place.

A checkpoint record can be left in the Checkpoint file if an interrupted Process is deleted by the operator. Connect:Direct scans the checkpoint records during initialization and deletes records older than the value specified in the CKPT.DAYS initialization parameter.

COPY Statement Parameters

The CKPT and REQUEUE parameters of the COPY statement also control aspects of the Checkpoint/Restart facility. The following table describes these parameters.

Parameter	Description
CKPT	Specifies the interval Connect:Direct uses to record checkpoint information. Connect:Direct uses this CKPT value, rounded to the nearest block, in determining how many bytes to transfer before taking a checkpoint. If you do not specify the Copy statement CKPT parameter, Connect:Direct uses the value specified in the CKPT initialization parameter. Specifying a value of CKPT=0K, or not specifying CKPT in the initialization parameters, disables checkpointing. Be aware of additional overhead associated with specifying too small a checkpoint interval, particularly when transferring large files.
REQUEUE	Indicates whether Connect:Direct requeues Processes that end due to an ABEND, such as an Sx37, or enables any subsequent steps to run to Process termination. This parameter is only effective if checkpointing is in use. See the Connect:Direct Processes Web site .

Checkpoint/Restart Examples for TCP/IP, UDT, or LU6.2 Transfers

The following examples describe how checkpoint/restart works for different transfer types.

TCP/IP, UDT, or LU6.2 Transfers

In the case of transfers using TCP/IP or UDT, Connect:Direct sends approximately 30 bytes of overhead after sending an amount of data equal to the checkpoint interval in effect. Therefore, if the file size is 3 million bytes and the checkpoint interval is 10k, then 30 bytes of overhead is sent 300 times, resulting in a total overhead of 9000 bytes.

Other Transfers

For other transfers (LU0 or transfers to MS-DOS), Connect:Direct sends an additional seven bytes per block or ten bytes per record of overhead during data transfer. Data is sent in record mode when reblocking is taking place during transfer (source and destination block sizes differ). When no reblocking is taking place, Connect:Direct sends in block mode.

When you enable checkpointing, Connect:Direct transfers positioning information in addition to the file. The positioning information enables Connect:Direct to reposition the file in the event of an interruption. The following examples show how to determine this overhead in number of bytes when using Checkpoint/Restart for non-TCP/IP transfers.

In the following table, Connect:Direct checkpoints the file in block mode and adds seven bytes to each block transmitted. This functionality adds only 2,100 bytes to the transmission of almost 6 million bytes.

File Attributes	Sending File	Receiving File
BLKSIZE	19,069	19,069
LRECL	0	0
DSORG	PS	PS

File Attributes	Sending File	Receiving File
RECFM	U	U
FILE SIZE	300 blocks	300 blocks

In the following table, Connect:Direct checkpoints the file in record mode because the data is being reblocked. Connect:Direct adds 10 bytes to each record being transmitted. This functionality adds 150,000 bytes to the transmission, or 5 percent overall.

File Attributes	Sending File	Receiving File
BLKSIZE	5,000	2,000
LRECL	200	200
DSORG	PS	PS
RECFM	FB	FB
FILE SIZE	600 blocks	1,500 blocks

RUN TASK Checkpoint/Restart

Connect:Direct for z/OS also offers a checkpoint/restart feature with the RUN TASK Process statement.

If a RUN TASK program is executing on the SNODE and a session failure occurs, the PNODE recognizes the session failure and puts the Process in the Timer queue for retry. The SNODE, however, is still running the RUN TASK program and is not aware of the session failure until the program finishes. The checkpoint/restart feature for RUN TASK ensures that when the Process restarts on the PNODE, the RUN TASK program does not execute a second time on the SNODE.

Checkpoint Records

Connect:Direct always writes a checkpoint record on the node where the RUN TASK program executes. The initial checkpoint record is written upon entry to DMGRUNT, the module that handles the RUN TASK Process statement. Connect:Direct updates the checkpoint record before attaching the program that is to execute. When the RUN TASK program finishes, Connect:Direct updates the checkpoint record again.

Restart Procedure

If a RUN TASK Process step restarts, the node where the program executes attempts to find the checkpoint record in the checkpoint file. If the RUN TASK step is still executing, the Process that is running for the restart of the step waits for the RUN TASK program to finish the first task and then proceed to the next step of the Process.

At Process restart for a RUN TASK step, if the program is still executing, you see two Processes in the EX queue for the same Process step. The first Process is executing the program. The second

Process is waiting for the first Process to complete. When the first Process completes, it determines that the session it was running under is lost and posts the second Process. The second Process records how the RUN TASK step that is still executing ended, and proceeds to the next step in the Process.

Re-execution of the RUN TASK

The RUN TASK step does not execute again if it is determined at restart that the RUN TASK step ended because it finished before the PNODE restarted the Process. However, if the RUN TASK program did not complete and is not currently running, then the RESTART parameter determines the restart of the Process.

Also, if at restart, Connect:Direct cannot find the checkpoint record and the RUN TASK program is not executing, Connect:Direct is unable to determine what action to take for the restart. If Connect:Direct cannot determine what action to take for the restart, it uses the RESTART parameter. You can code the RESTART parameter on the RUN TASK step or in the initialization parameters.

Note: If you code the RESTART parameter on the RUN TASK step, it overrides the initialization parameter. When you code RESTART=YES, Connect:Direct executes the program again. When you code RESTART=NO, the Process skips the RUN TASK step.

See the *Global Initialization Parameters* appendix of the *Connect:Direct for z/OS Administration Guide* for detailed information about the RUNTASK.RESTART initialization parameter.

Process Results and Statistics

This chapter describes the `SELECT STATISTICS` command which is used to retrieve statistics records. You can use this command to see the results of a Process that has finished executing.

Note: The Connect:Direct Activity Reporting System (ARS) produces detailed statistics reports. See the *Connect:Direct for z/OS Facilities Guide* for more information on this optional facility.

Note: You can also use the Connect:Direct Browser to perform some of the procedures in this chapter. To learn more about the Connect:Direct Browser, see the user's guide on the Connect:Direct Browser CD-ROM or available online from the Sterling Commerce Documentation Library.

Statistics Log Records

Connect:Direct keeps a record of operations and activity in a Statistics Log file. This file contains the following record types:

Record Type	Provides Statistics About
Process Submit	The submitted Process
Member Copy	Transmission of a PDS member
Copy	How the COPY statement completed
Process Duration	How the Process completed
Run Job	A job submitted to the z/OS internal reader (JES)
Run Task	A program attached to the Process as a subtask
Submit within Process	A Process that is submitted within another Process
InterConnect Option Results	Processes created by the InterConnect Option

For a complete table of statistics record types, refer to *Using Connect:Direct Exits* in the *Connect:Direct for z/OS Administration Guide*.

Statistics Records Content

The statistics records include the following information:

- ◆ In all records:
 - ◆ Record type or function
 - ◆ Date and Time of record logging
 - ◆ Process name and number
 - ◆ User ID
 - ◆ Other node (secondary or primary) involved in the Process
 - ◆ Start and stop time of the function
 - ◆ Highest or final completion code and message
 - ◆ Step label name, if one is specified
- ◆ In the Member Copy record, the input and output member names
- ◆ In the Process Submit and Submit Within records, the submitted PDS DSN and member name
- ◆ In the Copy record:
 - ◆ The number of records or blocks read from or written to the file, and whether the transfer was performed in block or record mode
 - ◆ The number of I/O bytes that were read or written from disk or tape
 - ◆ The number of VTAM bytes sent or received during the session, including Connect:Direct and VTAM control information
 - ◆ The RUSIZE

The size displayed is not the actual RUSIZE used by the transfer but rather the RUSIZE indicated by the LOGMODE. VTAM can negotiate for a smaller RUSIZE.
 - ◆ The compression percentage

This percentage is calculated by evaluating the number of VTAM bytes and the number of I/O bytes. If the number of VTAM bytes is less than or equal to the number of I/O bytes, a positive compression percentage is calculated as:

$$\text{Compression Percentage} = (1 - \{\text{VTAM Bytes} \div \text{I/O Bytes}\}) \times 100$$

If the number of VTAM bytes is greater than the number of I/O bytes, a negative compression is calculated as:

$$\text{Compression Percentage} = - (1 - \{ \text{I/O Bytes} \div \text{VTAM Bytes} \}) \times 100$$

The compression percentage values displayed in the Copy Termination record differ depending on the file attributes specified. A negative compression percentage means that more bytes are sent than are read from the file. This includes Connect:Direct control information.

Elapsed Time Significance

Use the following table to interpret the elapsed time between the start and stop times for the different record types:

Record Type	Interpretation
For Copy Termination	The elapsed time includes file allocation on both nodes and the time required to mount a tape, when required.
For Run Task or Run Job	The elapsed time indicates the time it took to run the task or job before control returned to Connect:Direct.
For Submit within Process	The elapsed time indicates the time it took to submit the Process to the TCQ.
For an ESF Process Submit	The start time is when the submit was performed (the Process was added to the TCQ). The stop time is when the Process was first added to the Process queue.

InterConnect Option Records

The InterConnect Option (ICO) routes files from a Connect:Direct node to a Connect:Enterprise node for distribution, distributes batches to a Direct node, and provides notification of success or failure of the Process. The InterConnect software copies the entire content of the SYSPRINT file of the Connect:Enterprise batch utility, STOUTL, into the Connect:Direct Statistics file.

This report from STOUTL is displayed as a block by Connect:Direct. The InterConnect Option software imposes a limit of 500 lines for the report. These reports are retrievable only through the SELECT STATISTICS command FILE option for batch, and the DISPLAY option for the IUI. See the *Offline Utilities* chapter of the *Connect:Enterprise for z/OS User's Guide* for samples of the output reports from the ADD and EXTRACT functions.

Additional statistics records related to the InterConnect Option are written to the statistics facility. For a list of those statistics records, refer to *Using Connect:Direct Exits* in the *Connect:Direct for z/OS Administration Guide*.

Note: If you specify a statistics exit in the initialization parameters, the LF and RO records are passed to the exit.

SELECT STATISTICS Command

Use the SELECT STATISTICS command to retrieve and examine statistic log records. You can select records based on certain conditions and indicate if you want the output displayed, printed, or saved in a file for later processing. See information about the FILE subparameter on page 24 for how to save your output in a file for processing later.

Command Format

SELECT STATISTICS has the following format and parameters. The required parameters and keywords are in bold print. Default values for parameters and subparameters are underlined.

Label	Command	Parameters
(optional)	SElect STATistics	WHERE (CCODE = (condition,completion code EXCLUDE = (MEMber MCR) (WTO) (NOTWTO) (list) MSGID = ID (list) PNAME = name (list) PNUMber = number (list) SERVER = server name STARTT = ([date day][,hh:mm:ssXM]) STOPT = ([date day][,hh:mm:ssXM]) USER = name (list) SNODE = name (list) TYPE = ID (list) FNAME = dsname (list) * CASE = YES NO ARCHDSN = dsname (list) FILE PRint TABLE SUMmary

The following table describes the SELECT STATISTICS command parameters:

Parameter	Description
WHERE CCODE = (condition,completion code EXCLUDE = (MEMber MCR) (WTO) (NOTWTO) (list) MSGID = ID (list) PNAME = name (list) PNUMber = number (list) SERVER = server name STARTT = ([date day][,hh:mm:ssXM]) STOPT = ([date day][,hh:mm:ssXM]) USER = name (list) SNODE = name (list) TYPE = ID (list) FNAME = dsname (list) * CASE = YES NO ARCHDSN = dsname (list))	<p>Specifies which statistics records to examine. This parameter is required. The subparameters, such as PNAME, PNUMBER, and STARTT, are optional, but you must include at least one subparameter.</p> <p>CCODE = (condition, completion code) specifies selection by completion code.</p> <p>condition specifies a relationship to the completion code given in the subsequent positional parameter.</p> <p>The options for specifying condition include:</p> <p>GT for greater than LT for less than EQ for equal to NE for not equal to GE for greater than or equal to LE for less than or equal to</p> <p>completion specifies a decimal value ranging from 1 to 2,147,483,647 to allow for all completion codes that the RUN TASK statement can pass. This last value represents a maximum 31-bit binary number.</p> <p>For example, if CCODE = (EQ,12) is specified, records that have a Comp Code of 0000000C are selected.</p> <p>For another example, if CCODE = (GT,0) is specified, you see statistics records in which the step completion code is greater than zero, as long as the records also meet other specified criteria.</p> <p>EXCLUDE = (MEMber MCR) (WTO) (NOTWTO) (list) specifies to exclude certain statistics from selection. Specify a list of excluded options by enclosing them in parentheses and separating them by a space or a comma. To select all statistics, omit the EXCLUDE parameter from the SELECT STATISTICS command. The TYPE parameter overrides the EXCLUDE specifications.</p> <p>MEMber MCR specifies whether or not to exclude the PDS member copy record for PDS copies on the statistics report.</p> <p>WTO specifies that Connect:Direct write-to-operator (WTO) messages are excluded from the SELECT STATISTICS command. Dynamic allocation messages are represented as WTOs in the statistics file.</p> <p>NOTWTO specifies that only WTO records are displayed (exclude everything that is NOT a WTO record).</p> <p>MSGID = ID (list) specifies selection by message ID. You can specify a list of message IDs by enclosing them in parentheses. You can use a generic specification in the MSGID parameter using an asterisk (*) at the end of the ID. For example, if you specify SCPA*, then MSGIDs with SCPA in the first four characters of the message ID are selected.</p>

Parameter	Description
WHERE (continued)	<p>PNAME = name (list) specifies selection by Process name. You can specify a list of Processes by enclosing them in parentheses. You can use a generic specification in the PNAME parameter by using an asterisk (*) at the end of the name. For example, if you specify PNAME=TEST*, then records with TEST in the first four characters of the Process name field are selected regardless of the contents of the remaining characters. Records having TEST, TEST123, and TESTX all satisfy this selection criterion.</p> <p>PNUMBER = number (list) specifies selection by Process number. You can specify a list of Processes by enclosing them in parentheses. The range is 1–199999.</p> <p>SERVER = server name specifies the name of the Connect:Direct member where the statistics are generated. The server name is a 1–8 character name assigned to each Connect:Direct Server through the CDPLEX.SERVER initialization parameter. This parameter is required in a Connect:Direct/Plex.</p> <p>STARTT = ([date day] [,hh:mm:ssXM]) specifies selection by designated starting date and time. The date or day and time are positional parameters. If you do not specify the date or day, precede the time with a comma.</p> <p>date specifies the date to execute the Process. You can specify the day (dd), month (mm), and year (yy for 2-digit year and yyyy for 4-digit year). You can use periods or backslashes (/) to separate the components of a date value. You can omit the separators only for transfers between mainframe nodes. Use separators to guarantee transfers between all platforms. You can use the following date formats, according to which date order is specified in the DATEFORM initialization parameter:</p> <ul style="list-style-type: none"> ◆ DATEFORM=MDY specifies the date format as mm/dd/yy, mm/dd/yyyy, mm.dd.yy, or mm.dd.yyyy ◆ DATEFORM=DMY specifies the date format as dd/mm/yy, dd/mm/yyyy, dd.mm.yy, or dd.mm.yyyy ◆ DATEFORM=YMD specifies the date format as yy/mm/dd, yyyy/mm/dd, yy.mm.dd, or yyyy.mm.dd ◆ DATEFORM=YDM specifies the date format as yy/dd/mm, yyyy/dd/mm, yy.dd.mm, or yyyy.dd.mm <p>Valid Julian dates formats are yyddd, yyyyddd, yy/ddd, yyyy/ddd, yy.ddd, or yyyy.ddd.</p> <p>If you do not specify the DATEFORM parameter, Connect:Direct for z/OS defaults to MDY date format.</p> <p>day specifies the day of the week to select. Valid names include MOnday, TUesday, WEdnesday, THursday, FRiday, SATurday, and SUnday. You can also specify YESTER to search for statistics records created yesterday or TODAY to search for the statistics records created today.</p>

Parameter	Description
WHERE (continued)	<p>hh:mm:ssXM indicates the time of day in hours (hh), minutes (mm), and seconds (ss) to select. Set XM to AM or PM. You can use the 24-hour clock or the 12-hour clock. If you use the 24-hour clock, valid times are 00:00-24:00. If you use the 12-hour clock, 1:00 hours are expressed as 1:00AM, and 13:00 hours are expressed as 1PM. If you use neither AM or PM, Connect:Direct assumes the 24-hour clock. You do not have to specify minutes and seconds.</p> <p>You can also specify NOON, which searches for the statistics records at noon, or MIDNIGHT, which searches for the statistics records at midnight. The time default is 00:00:00, the beginning of the day.</p> <p>If you specify time of day but not date, the output shows the first available entry in the statistics log for that time of day. Statistics from all later times and dates are output up to and including the stop time.</p> <p>If you do not specify the STARTT parameter, the search criterion consists of any time before the STOPT parameter.</p> <p>STOPT = ([date day] [,hh:mm:ssXM]) Specifies a search for statistics records up to and including the designated date, day, and time positional parameters. If you do not specify the date or day, place a comma before the time.</p> <p>The date and time formats are the same as the STARTT parameter. The time default is 24:00:00, the end of the day.</p> <p>If you specify time of day but not date, all statistics through the most recent entry for that time of day are output.</p> <p>If you do not specify the STOPT parameter, Connect:Direct selects statistics through the present time.</p> <p>USER = name (list) limits the selected statistics records to those that are written for users with the specified name. You can specify a list of names by enclosing them in parentheses. Use generic specifications by placing an asterisk (*) at the end of the name. For example, if you specify USER = SYS\$, then records with SYS\$ in the first four characters of the USER field are selected regardless of the contents of the remaining characters. Records having SYS\$BOB, SYS\$ADM, and SYS\$0001 all satisfy this selection criterion. USER names can be up to 64 characters in length and can contain lowercase characters.</p>

Parameter	Description																				
WHERE (continued)	<p>SNODE = name (list) limits the selected statistics records to those that are written for Processes where the specified node name acted as SNODE. You can specify a list of names by enclosing them in parentheses. Use generic specifications by placing an asterisk (*) at the end of the name. For example, if you specify SNODE=DALLAS*, then records with DALLAS in the first six characters of the SNODE field are selected regardless of the contents of the remaining characters. Records having DALLAS.PROD, DALLAS.TEST, and DALLAS all satisfy this selection criterion. SNODE names can contain lowercase characters.</p> <p>Note: When the statistics record shows NODE=S (SNODE), the PNODE and SNODE fields are switched in that Process' statistics record. For example, when a Run Task is submitted for SNODE processing, the RT statistics record created at the PNODE contains NODE=P and the PNODE and SNODE fields are as expected. For the same task, the RT statistics record created at the SNODE contains NODE=S and the PNODE and SNODE fields are reversed for that Process.</p> <p>TYPE = ID (list) * specifies the statistics record types to select. Every statistics record Connect:Direct generates has an associated record-type identifier. Each identifier is two characters long and indicates the event or function that generated the record. The identifier also indicates the record format and contents. Use TYPE= * to view all record types. The TYPE subparameter specifies which record types to select, and <i>overrides the EXCLUDE parameter</i>. When you do not specify a TYPE, the record types selected are determined by the output option chosen, and can be affected by the EXCLUDE subparameter. When you specify the PRINT or TABLE output option, the types in the following table are selected. Specify FILE for the output option to select all available types. Use the EXCLUDE subparameter to exclude certain types that the output option included.</p> <table border="1" data-bbox="773 1276 1122 1640"> <thead> <tr> <th data-bbox="773 1276 802 1304">ID</th> <th data-bbox="857 1276 992 1304">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="773 1314 802 1341">PS</td> <td data-bbox="857 1314 1024 1341">Process Submit</td> </tr> <tr> <td data-bbox="773 1352 802 1379">CT</td> <td data-bbox="857 1352 1040 1379">Copy Termination</td> </tr> <tr> <td data-bbox="773 1390 802 1417">MC</td> <td data-bbox="857 1390 1062 1417">PDS Member Copy</td> </tr> <tr> <td data-bbox="773 1428 802 1455">RT</td> <td data-bbox="857 1428 959 1455">Run Task</td> </tr> <tr> <td data-bbox="773 1465 802 1493">RJ</td> <td data-bbox="857 1465 943 1493">Run Job</td> </tr> <tr> <td data-bbox="773 1503 802 1530">SW</td> <td data-bbox="857 1503 1117 1530">Submit Within a Process</td> </tr> <tr> <td data-bbox="773 1541 802 1568">WO</td> <td data-bbox="857 1541 911 1568">WTO</td> </tr> <tr> <td data-bbox="773 1579 802 1606">PT</td> <td data-bbox="857 1579 1073 1606">Process Termination</td> </tr> <tr> <td data-bbox="773 1617 802 1644">TR</td> <td data-bbox="857 1617 1045 1644">SNMP Trap Event</td> </tr> </tbody> </table> <p data-bbox="773 1654 1419 1730">See <i>Using Connect:Direct Exits</i> in the <i>Connect:Direct for z/OS Administration Guide</i> for a list of all statistics record type identifiers.</p>	ID	Description	PS	Process Submit	CT	Copy Termination	MC	PDS Member Copy	RT	Run Task	RJ	Run Job	SW	Submit Within a Process	WO	WTO	PT	Process Termination	TR	SNMP Trap Event
ID	Description																				
PS	Process Submit																				
CT	Copy Termination																				
MC	PDS Member Copy																				
RT	Run Task																				
RJ	Run Job																				
SW	Submit Within a Process																				
WO	WTO																				
PT	Process Termination																				
TR	SNMP Trap Event																				

Parameter	Description
WHERE (continued)	<p>FNAME=dsname (list) * limits the selected statistics records to those that contain the specified file name. The FNAME subparameter is meaningful for the following record types:</p> <p>CT Copy Termination RJ Run Job SD Start Connect:Direct SW Submit within Process</p> <p>The meaning of the file name within these records is unique for each record type. For example, the Run Job record contains the file name of the submitted JCL. Filenames can be up to 254 characters in length and can contain lowercase characters.</p>
CASE=YES NO	<p>Specifies whether lowercase or mixed-case data is permitted for the USER, SNODE, and FNAME subparameters. The CASE subparameter overrides the global CASE option defined at signon for the purpose of the SELECT STATISTICS command.</p> <p>YES folds the data in USER, SNODE, and FNAME to uppercase regardless of the actual data specified.</p> <p>NO preserves the actual case typed for the USER, SNODE, and FNAME subparameters. The CASE defaults to the setting defined within the session defaults if nothing is specified.</p>
ARCHDSN=dsname (list)	<p>Specifies that Connect:Direct is to search the archived statistics files named, <i>instead of the statistics log currently in use by the DTF</i>. Archived statistics files store old statistics data that is no longer active in the statistics log. See your system administrator for information about the availability of archived statistics at your site, and for the data set names you can specify with the ARCHDSN subparameter. See page 156 for information on archiving statistics.</p>
FILE PRint TABle SUMmary	<p>Specifies the output format. Indicate only one. TABLE is the default. This parameter is optional.</p> <p>See page 24 for an explanation of the output produced by the selection of the FILE, PRINT, and TABLE parameters. The SUMMARY parameter produces a 3-line summary per statistics record, similar to the example on page 163.</p> <p>The reports from Connect:Enterprise are retrievable only through the FILE option for batch and the DISPLAY option for the IUI.</p>

Using SELECT STATISTICS through the Batch Interface

To use the SELECT STATISTICS command from the Batch Interface, perform the following steps:

1. Place commands in the DMBATCH job stream as described in Chapter 3, *The Batch Interface*.
2. Submit the job while Connect:Direct is running.

Command Examples

The following command searches for statistics records based on start and stop date and time, and excludes write-to-operator statistics messages.

```
SEL STAT WHERE ( -
  EXCLUDE= (WTO) -
  STARTT= (11/24/2003,06:45:00) -
  STOPT= (11/25/2003,8AM) -
)
```

The following command searches for statistics records based on start and stop date and time, and excludes write-to-operator and PDS member copy statistics records.

```
SEL STAT WHERE ( -
  EXCLUDE= (WTO,MEMB) -
  STARTT= (11/21/2003,15:25:00) -
  STOPT= (11/25/2003,15:30:00) -
)
```

The following command searches for statistics records based on Process name, and start and stop date and time. It excludes write-to-operator statistics messages.

```
SEL STAT WHERE ( -
  EXCLUDE= (WTO) -
  PNAME= BILLING -
  STARTT= (12/28/2003,06:00) -
  STOPT= (12/28/2003,23:00) -
)
```

The following command searches for signon records for the user Mary.

```
SEL STAT WHERE (USER=MARY, TYPE=SI) TABLE
```

The following command searches are identical and both select only WTO records.

```
SEL STAT WHERE (TYPE=WO) TABLE
SEL STAT WHERE (EXCLUDE= (NOTWTO)) TABLE
```

The following command searches for statistics records based on Process start/stop date and time that ran on a Connect:Direct Server named SERVER1.

```
SEL STAT WHERE ( -
                SERVER=SERVER1 -
                STARTT=(12/28/2003,06:00) -
                STOPT=(12/28/2003,23:00) -
                )
```

The following command searches for Process submission records for Processes in which the SNODE is any node in Dallas, and prints the output. Assume that the site administrator indicates that the November statistics records are contained in a file pair whose key sequenced data set is named SYS.ARCH.STATS.M9611. Refer to *Administering Statistics* in the *Connect:Direct for z/OS Administration Guide* for an explanation of statistics file pairs.

```
SEL STAT WHERE ( -
                TYPE=(PS,SW) -
                SNODE=DALLAS* -
                ARCHDSN=SYS.ARCH.STATS.M9611 -
                ) PRINT
```

The following command searches for signon and signoff records that match today's date and the User ID SEAN.

```
SEL STAT WHERE ( -
                TYPE=(SI,SO) -
                USER=SEAN -
                STARTT=(TODAY) -
                STOPT=(TODAY,24:00:00) -
                ) TABLE
```

The following command searches for the message ID SCMG010I and all message IDs that begin with SCPA.

```
SEL STAT WHERE ( -
                MSGID=(SCMG010I, SCPA*) -
                )
```

Sample Output

The following shows sample output from a SELECT STATISTICS command.

```

=====
node.name                SELECT STATISTICS                mm.dd.yyyy
=====
-----
Function      => PROCESS SUBMIT      Start Time => 15:21:11
Process Name  => BENCHRC              Stop Time  => 15:21:11
Process Num   => 272                Comp Code  => 00000000
                                           Comp Msg   => SSPA001I
Userid       => USER01
Primary      => node.name           Step Name  =>
Submitted DSN=> USER01.CD.PROCESS(SUB1)
-----
Function      => COPY                Start Time => 15:21:11
Process Name  => BENCHRC              Stop Time  => 15:21:12
Process Num   => 272                Comp Code  => 00000000
                                           Comp Msg   => SCPA000I
Userid       => USER01
Primary      => node.name           Step Name  => STEP1
Server       => S3                 PLEXCLASS => (          )
-----
From ( Snode
  Dsn=CSDQA1.TESTFILE.BENCH.M1)
  recs => 0                          blks => 36
  I/O BYTES => 1,000,000
  VTAM BYTES => 1,000,072
  Cmpr Perc => 0.0%
  Digital Signature enabled = No
  VOL=SER=> SVUS06
To ( Pnode
  Dsn=CSDQA1.FILETEST.BENCH01)
  recs => 0                          blks => 36
  I/O BYTES => 1,000,000
  VTAM BYTES => 1,000,072
  Cmpr Perc => 0.0%
  Digital Signature enabled = No
  Merged Signature enabled = No
  Merged Encryption enabled = No
  VOL=SER=> SVUS02
-----
Function      => Process Term      Start Time => 15:27:26
Process Name  => BENCHRC              Stop Time  => 15:27:30
Process Num   => 277                Comp Code  => 00000000
Server       => S2                 Comp Msg   => SVTM100I
Userid       => USER01
Secondary    => node.name
-----

```

The following figure illustrates the commands for archiving statistics for a single day using the Batch Interface. These commands archive the statistics from the previous day into the preallocated data set of the API referenced by the DD DMTMPDSN.

```
SIGNON      USERID=(USERID,PASSWORD) TMPDD=DMTMPDSN -
            NETMAP=HLQ.CD.NETMAP
SELECT STAT WHERE (STARTT=(YESTERDAY,00:00:00) -
                  STOPT=(YESTERDAY,24:00:00)) FILE
SIGNOFF
```

Using SELECT STATISTICS through the IUI

You can use the Select Statistics screen to select, display, and print statistics information from Connect:Direct activities.

1. Select option **SS** from the Primary Options Menu to display the Select Statistics screen

```
node.name          SELECT STATISTICS
CMD ==> P
Session not active. Only submits are allowed under ESF.          hh:mm
  CMD: S ... SUMMARY TABLE  D ... DISPLAY REPORT                yyyy.mm.dd
      P ... PRINT REPORT                                         yyyy.ddd

PROCESS NUMBERS:
  ==> _____  ==> _____  ==> _____  ==> _____

PROCESS NAMES:
  ==> _____  ==> _____  ==> _____  ==> _____

START DATE   ==> TODAY_____  ( YYYY.MM.DD )
START TIME   ==> _____  (HH:MM:SSXM)
STOP DATE    ==> TODAY_____  ( YYYY.MM.DD )
STOP TIME    ==> _____  (HH:MM:SSXM)

CONDITION CODE:  ==> __ _____

EXCLUDE ( MEMBER RECS ==> Y  WTO RECS ==> Y  ^WTO RECS ==> N )

CHANGE EXTENDED OPTS: ==> N
```

2. To access the Select Statistics Extended Options screen, type **Y** in the CHANGE EXTENDED OPTS field. On the command line, type the command for the output format that you want. The following table describes each option:

Command	Description
D	Display the output on the screen.
P	Send the output to the printer.
F	View unformatted statistic records.

Command	Description
S	Display a two-line summary per statistics record on the screen. See the Statistics Summary figure on page 163.

Note: The records from the InterConnect Option are only retrievable through the DISPLAY option.

The following table lists the statistics record types, their corresponding record type identifiers, and whether they display through the Display Report command, or the Change Extended Opts field of the Select Statistics screen.

Record ID	Description	Display Report	Change Extended Options
CE	Copy I/O Start	D	Y
CH	Change Process	D	Y
CI	Copy Step Start	D	Y
CS	Statistics Command		Y
CT	Copy Termination	D	Y
CX	Check Certificate Validity		Y
DC	Directory Commands	D	Y
DP	Delete Process	D	Y
DT	Select Task	D	Y
DU	Delete User		Y
EI	Event Services Start Command		Y
ET	Event Services Stop Command		Y
EV	Event Services Command		Y
FA	IGWFAMS Message		Y
FI	Long File Name Record		Y
FP	Flush Process	D	Y
FS	Suspend Process		Y
FT	Flush Task		Y
GO	Process Modal - GOTO, ELSE, or EXIT Statement		Y
HW	High Concurrent Session Count		Y

Record ID	Description	Display Report	Change Extended Options
IA	Inquire Statistics		Y
IB	Inquire Debug		Y
ID	Inquire STATDIR		Y
IF	Process Modal - IF Statement		Y
IK	Inquire APKey File		Y
IP	Inquire Initialization parameters		Y
IT	Inquire SNMP Trap Table		Y
IU	Insert User		Y
IX	Inquire Connect:Direct/Plex		Y
JI	Run Job Start		Y
LF	ICO Log File Record		Y
M2	Multiple Copy Record	D	Y
MC	PDS Member Copy	D	Y
NL	Process modal - EIF or PEND statement	D	Y
NM	NETMAP Updated		Y
PE	Connect:Direct/Plex Error Record		Y
PI	PNODE Process Start	D	Y
PR	Performance Measurement Record	D	Y
PS	Process Submit	D	Y
PT	PNODE Process Termination	D	Y
PX	Connect:Direct/Plex Activity (Leave or Join Connect:Direct/Plex)		Y
QE	Queue Change to EXEC Queue	D	Y
QH	Queue Change to HOLD Queue	D	Y
QP	Queue Change to Process Retention Queue	D	Y
QT	Queue Change to TIMER Queue	D	Y
QW	Queue Change to WAIT Queue	D	Y
RE	ICO Report Record		Y
RF	Refresh/Update initialization parameters		Y
RJ	Run Job		Y

Record ID	Description	Display Report	Change Extended Options
RO	ICO Event Record		Y
RT	Run Task		Y
S2	Statistics Logging Statistics	D	Y
SB	Session Begin	D	Y
SC	Statistics Control Record	D	Y
SD	Start Connect:Direct	D	Y
SF	Statistics Format		Y
SI	Signon	D	Y
SN	Select Netmap		Y
SO	Signoff	D	Y
SP	Select Process	D	Y
SS	Select Statistics	D	Y
ST	Stop Connect:Direct	D	Y
SU	Select User		Y
SW	Submit within a Process		Y
SY	SYSOPTS	D	Y
TF	TCQ Threshold Full		Y
TI	Run Task Start		Y
TL	TCQ Threshold Low		Y
TP	Throughput Record Statistics		Y
TR	Trap Event Record	D	
TS	Suspend Task	D	Y
TW	TCQ Threshold Warning		Y
UM	Update Network map		Y
UU	Update User		Y
VP	View Process	D	Y
WO	WTO	D	Y
WS	Select Stat Command		Y
XO	Trace On/Off	D	Y
ZI	SNODE Process Start	D	Y

Record ID	Description	Display Report	Change Extended Options
ZT	SNODE Process Terminated	D	Y

The following screen is displayed if you selected **Y** in the CHANGE EXTENDED OPTS field.

```

node.name          SELECT STATISTICS EXTENDED OPTIONS
CMD==> _____
                                                    hh:mm
                                                    yyyy.mm.dd
                                                    yyyy.ddd

RECORD             ==> __      ==> __      ==> __
TYPES:            ==> __      ==> __      ==> __

SNODE NAME:       ==> _____ MESSAGE ID:  ==> _____

USER ID:
  ==> _____

FILENAME:
  ==> _____

SEARCH            ==> _____
ARCHIVED          ==> _____
DATASETS:        ==> _____

SERVER:          ==> _____

DO YOU WANT THE VALUES OF THIS REQUEST TO BE CASE SENSITIVE?  ==> N

```

See the SELECT STATISTICS parameters description on page 150 for information on how to complete the fields, or press **PF1** for Help.

Statistics Summary

If you selected **S** to display a statistics summary, the following Statistics Summary screen is displayed.

```

-----STATISTICS SUMMARY-----
==>                                ROW 1 TO 3 OF 3
                                SCROLL ==> PAGE
FUNCTION  PNAME      PNUMBER      ----SUBMITTER NODE----OTHER.NODE-----
USERID
MSGID     RTNCD      END DATE/TIME  P |SNODE          SERVER
-----
*  RUN-TASK  TESTMBR2 5          SC.MVS.QA5A      AS400.CDQA62
   RBELL1
   ARTT003I 000000008 MM/DD/YY HH:MM:SS PNODE
S  COPY     TESTMBR2 5          SC.MVS.QA5A      AS400.CDQA62
   RBELL1
   SCPA001I 000000000 MM/DD/YY HH:MM:SS PNODE
   SUB-CMD  TESTMBR2 5          SC.MVS.QA5A      AS400.CDQA62
   RBELL1
   SSPA001I 000000000 MM/DD/YY HH:MM:SS PNODE

```

The first and second lines of each entry identify a Process by function, Process name and number, submitter node, other node, and user ID. The third line is the message ID, return code, ending date and time of that Process, and Connect:Direct Server that the Process ran on.

An asterisk (*) before the Process name indicates a nonzero return-code. You can type **M** next to the Process name to display a description of the message ID associated with the Process. Type **S** next to the Process name to show the statistics in detail, as in the following figure.

```

node.name          Connect:Direct (TM) MESSAGE DISPLAY          hh:mm
CMD ==>

MSGID ==> SVSL003I
MODULE ==> DMVSOPEN

Copy requested DISP=(,CATLG) to already cataloged dataset.

The PROCESS COPY step requested a DISP=(,CATLG) on the TO
clause of the COPY statement. The requested dataset already
exists as a cataloged dataset.

System Action. The PROCESS COPY step is terminated with a
completion code of 8.

Response: Either correct the COPY dataset or uncatalog the
existing dataset and re-submit the PROCESS.

```

Display Statistics

If you typed D to display the output on screen, it is formatted as shown in the following example.

Note: The statistics display function supports 133 character records and horizontal scrolling using the standard ISPF scrolling function keys.

```

=====
PLEX.JOE                                SELECT STATISTICS                                24.03.2003
=====
Function      => COPY                                Start Time      => 13:00:54
Process Name  => PRCDATIM                            Stop Time       => 13:00:55
Process Num   => 13273                               Comp Code       => 00000000
                                                    Comp Msg        => SCPA000I

Userid        => CBENN1
Secondary     => PLEX.TOM.TCP                        Step Name       => STEP2
Other addr    => 10.20.129.38
Other port    => 02038

                                                    V2 Buffer Size  => 65,536
Negotiated V2 Buffer Size => 65,536
TCP Buffer Size Used => 131,072

***** CHECKPOINTED; Interval = 1,000
From ( Pnode
  Dsn=CBENN1.V44.PDS)
  recs => 0                                blks => 2
  I/O BYTES => 8,400
  VTAM BYTES => 2,183
  Cmpr Perc => 74.0%                        CMPPrlevel=1    WINDOWsize=13    MEMlevel=4
  Digital Signature enabled = No
  VOL=SER=> USER19
To ( Snode
  Dsn=CBENN1.FTST.PRCDATIM.D1030325.T130042)
  recs => 0                                blks => 2
  I/O BYTES => 8,400
  VTAM BYTES => 2,183
  Cmpr Perc => 74.0%
  Digital Signature enabled = No
  Merged Signature enabled = No
  Merged Encryption enabled = No
  VOL=SER=> USER11

```

The following figure illustrates how output to the system console and messages in response to console commands are formatted.

```

BROWSE -- SYS.USER01.NDMAPI          ----- LINE 00000097 COL 001 080
COMMAND ==>>                          SCROLL ==>> CSR

-----
Function      => RUN TASK (PNODE)          Start Time   => 15:51:38
Process Name  => HYTADD01                  Stop Time    => 15:51:39
Process Num   => 261                      Comp Code    => 00000000
Server        =>                          Comp Msg     => SRTA000I
Userid        => USER01
SECONDARY     => USER01.MVS60.N2         STEP NAME    => MBA$CREL

15:51:39 SVTM052I MBA$CRELRUN TASK HYTADD01( 261) SNODE=USER01.MVS60.N2 (0
15:51:39 SDAA004I - LOCATE MSG=NO TSODSN=NO DSN=IOVER1.CMCDIO.PROFILES(SUBADDO
15:51:39 SDAC006I (LOCATE) - R15=0000, R0=0000, Function completed successfully.
15:51:40 SDAA004I - ALLOC TSODSN=NO,DD=NDM00017,RETURN=(DSORG,DEVTYPE),D
15:51:40 SDAB005I - ERR=0000, INFO=0000, DYNAMIC ALLOCATION COMPLETED SUCCESSFUL
15:51:40 SDAA004I - LOCATE MSG=NO TSODSN=NO DSN= USER01.STOUTL.SYSIN.T632272
=====
USER01.MVS60.N1                SELECT STATISTICS                DATE : 10/24/2003
=====
15:51:40 SDAC006I (LOCATE) - R15=0000, R0=0000, Function completed successfully.
15:51:40 SDAA004I - ALLOC TSODSN=NO,DD=NDM00018,RETURN=(DSORG,DEVTYPE),D
15:51:40 SDAB005I - ERR=0000, INFO=0000, DYNAMIC ALLOCATION COMPLETED SUCCESSFUL
15:51:41 SDAA004I - UNALLOC DD=NDM00017
15:51:41 SDAB005I - ERR=0000, INFO=0000, DYNAMIC ALLOCATION COMPLETED SUCCESSFUL

```

Note: In a Connect:Direct/Plex, all write-to-operator (WTO) records display the Connect:Direct Server name (or XCF.NAME for a Connect:Direct Manager) before the time.

SELECT MESSAGE Command

The SELECT MESSAGE command enables you to display message details using the message ID. The SELECT MESSAGE command has the following format and associated parameters. The required parameters and keywords are in bold print. Default values for parameters and subparameters are underlined.

Label	Command	Parameter
(optional)		SELect MeSsaGe WHERE (ID MSGID= message ID) [Short Long]

The following table describes the SELECT MESSAGE command parameters

Parameter	Description
WHERE ID MSGID = message ID	Specifies the 8-character message identification. You can specify a generic message ID.

Parameter	Description
[Short Long]	Specifies to display the message text in short or long format. The short format is the default.

The following screens show examples of the SELECT MESSAGE command.

To use SELECT MESSAGE with a specific ID:

```
SELECT MSG WHERE (MSGID=SCPA0001) LONG
```

To use SELECT MESSAGE with a generic ID:

```
SELECT MESSAGE WHERE (MSGID=SCPA*)
```

The Network Map

This chapter describes how to retrieve records from the network map file and translate TCP/IP host names to network addresses. For more information about the Connect:Direct network map, see *Maintaining the Network Map* in *Connect:Direct for z/OS Administration Guide*.

You can also use the Connect:Direct Browser to perform some of the procedures in this chapter. To learn more about the Connect:Direct Browser, see the user's guide on the Connect:Direct Browser CD-ROM or available online from the Sterling Commerce Documentation Library.

Retrieving Records from the Network Map File

Use the SELECT NETMAP command to retrieve records from the network map file for display or further processing. The network map file is maintained by your system administrator.

Issuing the SELECT NETMAP Command from the IUI

Use the Select network map screen to select a Network map record.

1. Select option **NM** from the Primary Options Menu to display the Select Netmap or TCP Information screen.

```

node.name                SELECT NETMAP OR TCP INFORMATION                hh:mm
CMD ==>

NETMAP INFORMATION

NODE KEY(S) : ==> _____                ==> _____
                ==> _____                ==> _____
                ==> _____                ==> _____
                ==> _____                ==> _____

TCP INFORMATION

ADDRESS: _____

NAME: _____

OUTPUT DESTINATION ==> DIS (DIS-DISPLAY, PR-PRINT)

```

2. Type the selection criteria for the nodes you want to examine and press **Enter**. To select all nodes, you can simply press **Enter**.

The following is an example of a network map showing two nodes: CD.PLX44 and CDSELECT.SCO.

```

BROWSE   SYS06097.T144239.RA000.USER01.NDMAPI.H02   Line 00000000 Col 001 080
Command ==>                                         Scroll ==> PAGE
***** Top of Data *****
=====
                SELECT NETWORK MAP
=====

Node Name   : CD.PLX44                VTAM Applid  : M1CDD97G
Max Parsess : 10                     Def Ses Class : 1
Session Type: SNA                     Environment   :
Node Status : INTERNAL, SEND, RECEIVE
ALT.COMM    : (ALT.DIR=BALANCE        -
              (ALT.ADDR=10.20.129.101,ALT.PORT=13641,
              ALT.TYPE=TCP , ALT.USE.OUT=YES ))
CRC         : DEFAULT                Session Snode Max : 255
PlexClass   : (                      ) Use.Server.Node: No
-----

Node Name   : CDSELECT.SCO           TCP PORT     : 13640
TCP Address : 10.20.246.244
Max Parsess : 20                     Def Ses Class : 2
Session Type: TCP                     Environment   : UNIX
Source IP   : 10.20.201.2
Node Status : INTERNAL, SEND, RECEIVE
ALT.COMM    : (ALT.DIR=TOP            -
              (ALT.ADDR=MVSLPARA.CSG.STERCOMM.COM,
              ALT.PORT=01364,ALT.TYPE=TCP ,
              SOURCEIP=10.20.202.22 ,
              ALT.USE.OUT=YES ))
CRC         : DEFAULT                Session Snode Max : 255
Contact     : Joe Dowe
Phone       : (xxx) xxx-xxxxx
Description : Connect:Direct for z/OS XXXX Node

```

You can make generic requests by using an asterisk. For example, type an asterisk (*) in the first node key field to select all nodes and **D*** to select only those nodes that begin with **D**. See *SELECT NETMAP Command Format* on page 172 or press **PF1** for Help.

Note: The TCP information fields are for creating the SELECT TCPXLAT command described in the next section. Do not fill them in for the SELECT NETMAP command.

Issuing the SELECT NETMAP Command from the Batch Interface

To use the SELECT NETMAP command from the Batch Interface, place your commands in the DMBATCH job stream and submit the job while Connect:Direct is running. The following command displays all network map entries for node names that begin with *SAN*.

```

SIGNON
SEL NET WHERE (NODE=(SAN*))
SIGNOFF

```

In the following example, the only node name in the network map which begins with SAN is the SANFRAN node. This report contains a long DNS record, which you can define in the Adjacent Node Definition in the network map. For more information on the Adjacent Node Definition, refer to *Maintaining the Network Map* in *Connect:Direct for z/OS Administration Guide*.

```

=====
                        SELECT NETWORK MAP
=====
NODE NAME   : SANFRAN           VTAM APPLID   : M1A66789
MAX PARSESS : 10                DEF SES CLASS : 1
SESSION TYPE: SNA                ENVIRONMENT   : OS390
NODE STATUS : INTERNAL, SEND, RECEIVE
LONG DNS    : REGIONAL.MANAGER.OFFICE.IN.THE.SAN.FRANCIS
              CO.BAY.AREA.CALIFORNIA.USA
ALT.COMM    : (ALT.DIR=BALANCE   -
              (ALT.ADDR=M1A66790,ALT.TYPE=SNA  ,
              ALT.USE.OUT=YES ))
CRC         : DEFAULT           SESSION SNODE MAX : 255
CONTACT    : JOE SMITH
PHONE     : 345-999-6999
DESCRIPTION : SANFRAN TO TOKYO NODES PATH,  SECONDARY P
              ATH TO OSAKA.

```

SELECT NETMAP Command Format

The SELECT NETMAP command uses the following format and parameters. The required parameters are in bold print. Default values for parameters and subparameters are underlined>.

Label	Command	Parameters
(optional)	SElect NETMAP	WHERE (NODE = (node generic [(list))) PRint DISplay

The following table describes SELECT NETMAP parameters:

Parameter	Description
WHERE (NODE = (node generic (list)))	<p>Specifies which network map node definitions you want to examine. WHERE is the only required parameter for the SELECT NETMAP command.</p> <p>NODE=(node generic (list)) specifies the network map node names that are selected. Specify a list of node names by enclosing them in parentheses and separating each by a comma or a blank.</p> <p>node specifies a node name. The node is a 1–16 character alphanumeric string, with the first character alphabetic.</p> <p>generic specifies a generic selection of node names. To specify node names generically, type a 1–7 character alphanumeric string, with the first character alphabetic, plus an asterisk (*). For instance, if your network includes node names PHOENIX, SANDIEGO, SANFRAN, and TUCSON, a specification of SAN* provides information about the SANDIEGO and SANFRAN nodes. If you type only an asterisk (*), Connect:Direct displays (or prints) all nodes that you are authorized to use.</p>
PRint <u>DI</u> splay	Indicates the output format. Indicate only one.

Translating TCP/IP Host Names to Network Addresses

Use the SELECT TCPXLAT command to translate TCP/IP host names to network addresses and TCP/IP addresses to host names.

Issuing the SELECT TCPXLAT Command from the IUI

The Select NETMAP or TCP Information screen issues the SELECT TCPXLAT command or the SELECT NETMAP command depending on your request.

1. Select option **NM** from the Connect:Direct Primary Options Menu to display the Select Netmap or TCP Information screen. (See *Issuing the SELECT NETMAP Command from the IUI* on page 169 to see a screen sample.)
2. Press **Tab** until you reach the TCP INFORMATION portion of the screen. See the parameter descriptions on page 175, or press **PF1** for Help.
3. Do one of the following:
 - ◆ To translate a host address to a fully qualified host name, type the address in dot notation form in the ADDRESS field.
 - ◆ To translate a host name to a real TCP/IP address, type the host name in the NAME field. You can also type an alias name in this field if the host name is too long for the space provided.

The following output is displayed:

```

CD.DEV.STHOM1          RESOLVED TCP NAME/ADDRESS          14:56
CMD ==>
  TCP name/address translation successful.

      TCP ADDRESS : 10.20.42.192

      FULLY QUALIFIED TCP HOST NAME:

      qaoptsol.csg.stercomm.com

      _____ $$$$ _____

```

Issuing the SELECT TCPXLAT Command with the Batch Interface

To use the SELECT TCPXLAT command from the Batch Interface, place your commands in the DMBATCH job stream and submit the job while Connect:Direct is running. Use the batch version of the SELECT TCPXLAT command to make multiple requests for resolution. The maximum number of requests is 10.

The following command produces the TCP Name/Address Translation Report. The first two names listed in the TRTCPNAM parameter are aliases, the third is an actual host name.

```

SIGNON
  SEL TCPXLAT WHERE (TRTCPNAM=(NAME1, NAME2, tcpip.host.name3))
  SEL TCPXLAT WHERE (TRTCPADR=(123.456.78.999, 123.456.789.012))
SIGNOFF

```

The following report shows the resolution of name to address or address to name. REQUEST is what is typed in the command.

```

=====
                        TCP Name/Address Translation Report
=====
REQUEST                  : NAME1
Host Address              : NNN.NNN.NN.NNN..
HOST NAME                 : tcpip.host.name1

REQUEST                  : NAME2
*** TCP Name/Address could not be resolved ***

REQUEST                  : tcpip.host.name3
Host Address              : NNN.NNN.NN.NNN..
HOST NAME                 : tcpip.host.name3
=====
                        TCP Name/Address Translation Report
=====
REQUEST                  : 123.456.78.999
Host Address              : 123.456.78.999,
HOST NAME                 : tcpip.host.name4

REQUEST                  : 123.456.789.012
Host Address              : 123.456.789.012,
HOST NAME                 : tcpip.host.name5

```

SELECT TCPXLAT Command Format

The SELECT TCPXLAT command uses the following format and parameters. The required parameter is in bold print. Default values for parameters and subparameters are underlined.

Label	Command	Parameters
(optional)	SElect TCPXLAT	WHERE (TRTCPNAM = (tcpip.host.name (list)) TRTCPADR = (tcp.net.adr (list))) PRint DISplay

The following table describes the SELECT TCPXLAT command parameters:

Parameter	Description
WHERE (TRTCPNAM = (tcpip.host.name (list)) TRTCPADR = (tcp.net.adr (list)))	Specifies which TCP host name or network address resolution you want to examine. This parameter is required. Provide one of the following subparameters: TRTCPNAM = (tcpip.host.name (list)) specifies a translation from host name or alias to a real address. Type this name in the standard format with each of the qualifiers being from 1 to 64 alphanumeric characters, with a maximum name length of 64 characters. Specify a list of names by enclosing them in parentheses and separating each by a comma or a blank. TRTCPADR = (tcp.net.adr (list)) specifies a translation from an IP address to the host name for that network. Type this address in the dotted format (nnn.nnn.nnn.nnn) with a maximum length of 15 characters including periods. Specify a list of addresses by enclosing them in parentheses and separating each by a comma or a blank.
PRint DISplay	Specifies where output is directed. Indicate only one. When you submit this command through DMBATCH, use the PRINT parameter only.

Utility Programs

Connect:Direct provides several utility programs and one I/O exit that facilitate your use of the software. Use the information in this chapter to perform the following functions:

- ◆ Notify the user of Process success or failure using DMNOTIFY and DMNOTFY2
- ◆ Dynamically allocate data sets using DMRTDYN
- ◆ Place job stream in a wait status for a specified period of time and set return codes indicating whether the batch step or RUN TASK completed successfully or failed by using the DMRTWAIT program. You can use DMRTWAIT with the MAXDELAY parameter in a submit command or PROCESS statement to synchronize Processes submitted through the batch interface.
- ◆ Dynamically invoke AMS (Access Method Services) to perform VSAM utility functions using DMRTAMS
- ◆ Resolve symbolic references using DMRTSUB and RTGDGSUB
- ◆ Compress and decompress files stored in a ZLIB compressed format (CDSACOMP)
- ◆ Determine how many concurrent sessions are running during a specific time period (DMRTVITL)
- ◆ Interface with the IBM utility ADRDSSU, by using the Connect:Direct I/O exit, DMDSSIOX. The ADRDSSU program enables you to copy SMS-compressed data without having to decompress the data and also provides support for copying wildcard-named files.
- ◆ Replace existing FTP sessions within z/OS job streams with Connect:Direct for z/OS Processes using Connect:Direct FTP+ for z/OS (CDFTP).

User Notification Programs

Connect:Direct provides sample programs that notify the console operator or IUI users about the success or failure of any Process step. The sample programs are supplied in source statement form in the Connect:Direct SAMPLIB and in load module form in Connect:Direct LINKLIB. Use RUN TASK and MODAL statements with the sample programs to notify users.

The messages sent are standard TRANSFER SUCCESSFUL or TRANSFER FAILED status messages.

Operator Console Notify Program (DMNOTIFY)

The operator console program, DMNOTIFY, works with the RUN TASK statement to notify the console operator of the success or failure of Processes by means of Write to Operator (WTO) messages.

The program must have the following parameters passed to it:

Parameter	Description
CL4'GOOD' CL4'FAIL'	The first positional parameter tells the program the correct message to output. Code the parameter as illustrated in the following example, including the single quotes.
file name	The second positional parameter is the name of the file that is being copied.

The following example shows a Connect:Direct for z/OS Process using the DMNOTIFY program.

```

COPY1 PROCESS PNODE=CDA SNODE=CDB
STEP01 COPY -
        FROM ( -
                PNODE -
                DSN=FILE1 -
                DISP=SHR -
                ) -
        TO ( -
                SNODE -
                DSN=FILE2 -
                DISP=SHR -
                )
STEP02 IF (STEP01=0) THEN
NOTIFYG RUN TASK (PGM=DMNOTIFY, -
                PARM=(CL4'GOOD', FILE1)) PNODE
        ELSE
NOTIFYF RUN TASK (PGM=DMNOTIFY, -
                PARM=(CL4'FAIL', FILE1)) PNODE
EIF

```

The DMNOTIFY program is attached on the PNODE. A parameter list containing the GOOD or FAIL message criteria and the FILE1 file name is passed to DMNOTIFY. If the COPY statement is successful, the first RUN TASK step executes and you see the following message.

```

SNTB001I ** Connect:Direct SUCCESSFUL STEP COMPLETION **
FILE XFER BETWEEN - PLEX.JOE AND - PLEX.TOM SUCCESSFUL
REQUESTOR=CTENN1
DSN=CD.OUTPUT.data set

```

If the COPY statement is unsuccessful, the second RUN TASK step executes and you see the following messages.

```

SNTB002I ** Connect:Direct FAILED STEP COMPLETION **
***** Connect:Direct FILE TRANSFER FAILED *****
***** Connect:Direct FILE TRANSFER FAILED *****

FILE XFER BETWEEN - PLEX.JOE AND - PLEX.TOM FAILED
REQUESTOR=CTENN1
DSN=CD.OUTPUT.data set

***** Connect:Direct FILE TRANSFER FAILED *****
***** Connect:Direct FILE TRANSFER FAILED *****

```

User Notify Program (DMNOTFY2)

Use the sample program, DMNOTFY2, in conjunction with the RUN TASK statement to notify TSO users of various conditions. The program accepts a variable length parameter list to broadcast notification messages to users.

Note: The TSO user ID must be in the same z/OS image as the DTF where the DMNOTFY2 RUN TASK is running.

The following table describes the parameters:

Parameter	Description
CL4'GOOD' CL4'FAIL'	This first positional parameter indicates the success or failure of the Process step. Code the parameter as illustrated in the following example, including the single quotes.
file name	This second positional parameter is the name of the file that is being copied.
user ID user ID list	The last parameter or list of parameters contains the TSO user IDs to notify. Separate the user IDs with a comma.

The following example demonstrates a Process using the DMNOTFY2 program.

```

COPY PROCESS PNODE=CDA SNODE=CDB
STEP01COPY -
    FROM (PNODE -
          DSN=FILE1 -
          DISP=SHR) -
    TO (SNODE -
        DSN=FILE2 -
        DISP=RPL )
STEP02 IF (STEP01=0) THEN
NOTIFY1 RUN TASK -
        (PGM=DMNOTFY2, PARM=(CL4'GOOD', FILE1, CDID1, CDID2)) SNODE
    ELSE
NOTIFY2 RUN TASK -
        (PGM=DMNOTFY2, PARM=(CL4'FAIL', FILE1, CDID1, CDID2)) PNODE
EIF

```

A GOOD value is relayed to the TSO users CDID1 and CDID2 on CDB in the following message.

```
SNTA001I **Connect:Direct FILE TRANSFER SUCCESSFUL**
      TRANSFER BETWEEN - CDA  AND  - CDB
      DSN=FILE1
```

A FAIL value is relayed to the TSO users CDID1 and CDID2 on CDA in the following message.

```
SNTA002I ****Connect:Direct FILE TRANSFER FAILED****
      TRANSFER BETWEEN - CDA  AND  - CDB
      DSN=FILE1
```

Dynamic Allocation Program (DMRTDYN)

The dynamic allocation program, DMRTDYN, performs several useful status checking and housekeeping functions in a Process. It is located in the Connect:Direct LINKLIB.

When multiple actions are contained in the parameters, the completion code returned is the highest completion code encountered.

Parameters

The RUN TASK command invokes DMRTDYN with parameters that define allocation actions to execute. The following rules apply to the syntax of the parameters:

- ◆ Perform each allocation action with as many parameters as required.
- ◆ Separate each unique allocation action by a parameter containing a fullword of minus one (F²-1').
- ◆ A leading blank on the first parameter is not required; however, you must include a leading blank on all subsequent parameters. If you do not include a leading blank, the program issues an RC=30 and an SRTA005I message for the Process.
- ◆ Parameters must not extend past column 72, or they are not read.

The following table shows the available dynamic allocation functions.

Function	Description
ALLOC	Use the ALLOC function to allocate a file. The allocation request can contain most job stream DD statement options, including DCB options applicable to allocation. This function is the default.
UNALLOC	Use the UNALLOC function to unallocate a file.
CONCAT	Use the CONCAT function to concatenate two files.
DECONCAT	Use the DECONCAT function to perform dynamic deconcatenation.
LOCATE	Use the LOCATE function to locate cataloged data sets.

Note: You cannot access HFS files using DMRTDYN.

Sample Program Uses

The following examples demonstrate how you can use the functions of the DMRTDYN program.

Example 1: Locate and Allocate a Data Set

The following example calls the program DMRTDYN using the LOCATE parameter to determine if a cataloged file exists.

If the file does not exist, the program issues a nonzero return code and makes a call to allocate the file through the ALLOC parameter.

```

PROC1 PROCESS SNODE=BOSTON
LOCATE    RUN TASK (PGM=DMRTDYN -
           PARM=(C'LOCATE DSN=XYZ.CUSTOMER.FILE')) SNODE
           IF (LOCATE NE 0) THEN
ALLOC     RUN TASK (PGM=DMRTDYN -
           PARM=(C'ALLOC', -
           C' DSN=XYZ.CUSTOMER.FILE', -
           C' DISP=(NEW,CATLG)', -
           C' VOL=SER=DCM009', -
           C' SPACE=(CYL,(1,1))', -
           C' DSNTYPE=EXTPREF', -
           C' DCB=(RECFM=FB,DSORG=PS,LRECL=80,BLKSIZE=3120)', -
           C' UNIT=3380')) SNODE
EIF

```

Example 2: Delete and Uncatalog a Data Set

The following example deletes and uncatalogs a data set.

The first function, ALLOC, allocates the data set specifying a current status of OLD and a secondary (or normal step completion) disposition of DELETE, which deletes the data set.

The second function, UNALLOC, unallocates the data set. Because of the disposition (the data set is originally allocated with disposition DELETE), the data set is uncataloged.

Use the F'-1' as a separator between allocation functions.

```

PROC2 PROCESS SNODE=BOSTON
DELETE RUN TASK (PGM = DMRTDYN -
           PARM = (C'ALLOC DSN=XYZ.TEMP DISP=(OLD,DELETE)' -
           F'-1' -
           C'UNALLOC DSN=XYZ.TEMP'))

```

Example 3: Allocate and Concatenate Data Sets

The following example calls the program to allocate each data set through the ALLOC parameter. Then it calls DMRTDYN again to do the concatenation through the CONCAT parameter. The data set is concatenated under the first DDNAME in the CONCAT list.

```

RTDYNCON PROCESS SNODE=BOSTON
ALLOC1 RUN TASK (PGM=DMRTDYN -
  PARM=(C'ALLOC DSN=XYZ.TEMP DISP=SHR DD=OLD1')) PNODE
ALLOC2 RUN TASK (PGM=DMRTDYN -
  PARM=(C'ALLOC DSN=XYZ.TEMP2 DISP=SHR DD=OLD2')) PNODE
CONCAT RUN TASK (PGM=DMRTDYN -
  PARM=(C'CONCAT DD=(OLD1,OLD2)')) PNODE

```

Example 4: Deconcatenate and Unallocate Data Sets

The following example calls the program to deconcatenate by DDNAME using the DECONCAT parameter. Then it calls DMRTDYN again to unallocate each data set in the concatenation with the UNALLOC command.

```

RTDYNCON PROCESS SNODE=BOSTON
DECONCAT RUN TASK (PGM=DMRTDYN -
  PARM=(C'DECONCAT DD=(OLD1)')) PNODE
UNALLO1 RUN TASK (PGM=DMRTDYN -
  PARM=(C'UNALLOC DD=OLD1')) PNODE
UNALLO2 RUN TASK (PGM=DMRTDYN -
  PARM=(C'UNALLOC DD=OLD2')) PNODE

```

Run Task Wait Program (DMRTWAIT)

Use the DMRTWAIT program as a batch step or a RUN TASK to place a job stream in a wait status for a specified period of time and set return codes indicating when the step or task completes. To synchronize Processes submitted through the batch interface, use DMRTWAIT with the MAXDELAY parameter in a SUBMIT command or PROCESS statement. For more information, see *Using the MAXDELAY Keyword Parameter* on page 29. This program is distributed via SAMPLIB and must be assembled and link-edited to the Connect:Direct load library.

The following table describes the positional parameters. If you execute DMRTWAIT as a RUN TASK, enclose the parameters in quotes because of the embedded commas, for example, PARM=('1,2').

Parameter	Description
VALUE1	The first positional parameter indicates the time to wait. The default is 0 or you can use a comma to omit. For seconds, specify (S)xxxx. The maximum is 9999 seconds or about 2.7 hours. For minutes, specify Mxxx. The maximum is 999 minutes or about 16.6 hours. For hours, specify Hxx. The maximum is H99 or about 4.1 days. You can also specify 'ABEND' and then use the actual user ABEND code for the second positional parameter.
VALUE2	The second positional parameter specifies the return code to set. The default is 0 The maximum is 4095. '-' causes a -1 value (if executed as a batch step, IBM displays the return code value as 4095).
VALUE3	The third positional parameter is only used to display diagnostics as WTO (write-to-operator) messages. Specify 'DIAG' or leave blank. Any other value is ignored.

The following examples show how to use DMRTWAIT in a RUN TASK. In the first example, the program waits for one second, and then sets the return code to 2.

```
DMRTWAIT PROCESS SNODE=NODE.NAME &PRM='1,2'
STEP01 RUN TASK (PGM=DMRTWAIT PARM=(&PRM))
```

In this example, the program does not wait, but sets the return code to 8.

```
DMRTWAIT PROCESS SNODE=NODE.NAME &PRM='0,8'
STEP01 RUN TASK (PGM=DMRTWAIT PARM=(&PRM))
```

VSAM AMS Interface Program (DMRTAMS)

The VSAM AMS interface program, DMRTAMS, dynamically invokes AMS to perform VSAM utility functions. The program is located in the Connect:Direct LINKLIB.

The highest completion code encountered in any AMS action is returned.

Parameters

DMRTAMS is invoked by the RUN TASK Process statement with the following parameters that define allocation and AMS actions to execute.

Parameter	Description
sysprint output parameters	<p>The first positional parameter is special because it defines dynamic allocation parameters for the AMS SYSPRINT output. This parameter can allocate a spool file or a user file. The FREE and RETURN subparameters are required.</p> <p>Note: You must specify all of the sysprint output parameters on the same line. They cannot be on separate lines.</p> <p>The following example defines a CLASS A SYSOUT spool file.</p> <pre>C'FREE=CLOSE, RETURN=(DD) SYSOUT=A'</pre>
control statement parameters	<p>You can use any valid input to the VSAM AMS program as input to DMRTAMS, even multiple input parameters defining one or more AMS actions. You must begin all VSAM control statement parameters with a blank.</p>

Sample Program

The following figure shows a sample of DMRTAMS using the Define Cluster AMS function. The ASM SYSPRINT output is routed to the existing user file XYZ.OUTPUT.

```
VADC1 PROCESS SNODE=BOSTON
DEFINE RUN TASK (PGM=DMRTAMS, -
  PARM=(C'FREE=CLOSE,RETURN=(DD),DISP=OLD,DSN=XYZ.OUTPUT', -
    C' DELETE(VSAM.PAY.CHECKS) CLUSTER', -
    C' DEFINE CLUSTER -',-
    C' (NAME(VSAM.PAY.CHECKS) -',-
    C' RECORDS(1) -',-
    C' VOLUMES(DCM009) -',-
    C' OWNER(NDM) -',-
    C' NONINDEXED -',-
    C' RECORDSIZE(4089 4089) -',-
    C' SHAREOPTIONS(2)) -',-
    C' DATA -',-
    C' (CONTROLINTERVALSIZE(4096) -',-
    C' NAME(VSAM.PAY.CHECKS.DATA)) -',-
    C' CATALOG(USER.UCAT')) SNODE
```

Symbolic Resolution Utilities (DMRTSUB and RTGDGSUB)

The Connect:Direct utility programs, DMRTSUB and RTGDGSUB, enable you to submit jobs to the internal reader, much like RUN JOB. DMRTSUB has the added capability of symbolic resolution into the job. The RUN TASK statement invokes DMRTSUB or RTGDGSUB, which allows you to substitute variables in the parameter list.

The following table lists DMRTSUB and RTGDGSUB parameters, and RTGDGSUB variables:

Parameter	Description								
JCL source	The first parameter is an allocation string to point to the job to submit. Code it without a ddname because DMRTSUB acquires one. Connect:Direct uses this value, along with the DISP parameter, to allocate the data set that contains the JCL.								
Subsequent Parameters	<p>Subsequent parameters define resolutions for symbols used in the JCL itself. For each parameter, the first position is the name of the symbol found in the JCL and the second position is the value used. Separate the first and second positions by an equal sign (=), one or more spaces, or a comma. The value is terminated by the end of the parameter string, or a comma or space not within a quoted string. A parameter can also contain only a symbol name, in which case the value to substitute into the job stream is a null string.</p> <p>Note: To pass lowercase data to the DMRTSUB program, use SYSOPTS in the RUN TASK statement.</p>								
Variables (for use with RTGDGSUB)	<table> <tbody> <tr> <td>%SRCDSN</td> <td>The resolved 'FROM' data set name from the last COPY step executed in this Process</td> </tr> <tr> <td>%DSTDSN</td> <td>The resolved 'TO' data set name from the last COPY step executed in this Process</td> </tr> <tr> <td>%PROCNAME</td> <td>The name of this Process</td> </tr> <tr> <td>%PROCNUM</td> <td>The number of this Process</td> </tr> </tbody> </table> <p>Note: To use %SRCDSN or %DSTDSN, the RUN TASK must be run on the PNODE. The values for the %SRCDSN and %DSTDSN variables are not available when the RUN TASK is run on the SNODE.</p>	%SRCDSN	The resolved 'FROM' data set name from the last COPY step executed in this Process	%DSTDSN	The resolved 'TO' data set name from the last COPY step executed in this Process	%PROCNAME	The name of this Process	%PROCNUM	The number of this Process
%SRCDSN	The resolved 'FROM' data set name from the last COPY step executed in this Process								
%DSTDSN	The resolved 'TO' data set name from the last COPY step executed in this Process								
%PROCNAME	The name of this Process								
%PROCNUM	The number of this Process								

The following figure is an example of RUN TASK using the DMRTSUB program.

```

RUN TASK (PGM=DMRTSUB -
          PARM=(' DSN=dsname,DISP=SHR', -
              ' JOB X21JOB', -
              ' RPTNAME REPORT01', -
              ' CLASS A', -
              ' FILE X21.FILE' -
              )) PNODE

```

The job stream source that corresponds to this RUN TASK statement is displayed in the following figure. The values requiring substitution are presented in bold type.

```

//&JOB JOB (79502), &RPTNAME, PRTY=12, TIME=(1), CLASS=&CLASS,
// REGION=512K, MSGLEVEL=(1,1), MSGCLASS=X, NOTIFY=BSMITH1
//PRINT EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY, DCB=BLKSIZE=80
//SYSUT1 DD DISP=SHR,
// DSN=&FILE
//SYSUT2 DD SYSOUT=&CLASS, DCB=( &FILE)

```

The following figure shows the resulting job stream after substitution has taken place. Connect:Direct replaces the variables that began with & with the assigned values in the DMRTSUB parameter list.

```

//X21JOB JOB (79502), REPORT01, PRTY=12, TIME=(1), CLASS=A,
// REGION=512K, MSGLEVEL=(1,1), MSGCLASS=X, NOTIFY=BSMITH1
//PRINT EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY, DCB=BLKSIZE=80
//SYSUT1 DD DISP=SHR,
// DSN=X21.FILE
//SYSUT2 DD SYSOUT=A, DCB=(X21.FILE)
//*

```

Passing Variables Using RTGDGSUB

You can use the RTGDGSUB program, which substitutes variables in the parameter list and then calls the DMRTSUB program to process the parameter list further.

To do this, modify an existing RUN TASK statement that calls DMRTSUB to call RTGDGSUB, and add the variables you want to use in the parameter list.

The following figure is an example of RUN TASK using the RTGDGSUB program.

```

STEP01 RUN TASK(PGM=RTGDGSUB -
  PARM=('DSN=DALLAS1.ZOS.SRCLIB(JCLTEST1), DISP=SHR', -
    'JOB &JOBNAME', -
    'SRC %SRCDSN', -
    'DST %DSTDSN', -
    'PROCNAME %PROCNAME', -
    'PROCNUM %PROCNUM', -
    'DSN DSNNAME')) PNODE

```

In this example, the following circumstances are in effect:

- ◆ The 'FROM' data set in the last COPY step before this RUN TASK is DALLAS1.FILE
- ◆ The 'TO' data set in the last COPY step before this RUN TASK is DALLAS1.TEST.GDG.G0006V00
- ◆ The name of the Process is 'TESTPR'
- ◆ The Process number is 29
- ◆ The &JOBNAME is XYZ

The RTGDGSUB program makes the variable substitutions and calls the DMRTSUB program with the following parameters:

```

    PARM= (' DSN=DALLAS1.ZOS.SRCLIB(JCLTEST1), DISP=SHR', -
          'JOB XYZ', -
          'SRC DALLAS1.FILE', -
          'DST      DALLAS1.TEST.GDG.G0006V00    ', -
          'PROCNAME TESTPR', -
          'PROCNUM 00029 ', -
          'DSN  DSNNAME'))
  
```

Wherever &JOB, &SRC, &DST, &PROCNAME, &PROCNUM, and &DSN are found in DALLAS1.ZOS.SRCLIB(JCLTEST1), the corresponding values are substituted. Once all the substitutions have been made, the resultant JCL is submitted to JES.

Using Symbolic Variables

It can also be useful to specify the values to substitute as symbols rather than their actual value. You can then use a single Process for multiple purposes.

The following figure is an example of DMRTSUB that symbolically substitutes values into a z/OS job being submitted to the internal reader. Connect:Direct supports any symbols that begin with an ampersand.

In the example, the second and subsequent parameters name the symbol, without the &, and value that the symbol becomes in the submitted job stream. Connect:Direct substitutes the value itself when the Process is submitted.

```

    DMRTSUB PROC SNODE=CD.OS39040.N1 NOTIFY=USER01 -
            &CLASS=A &DEST=LOCAL &JOB=JOB1SB -
            &RPTNAME='''RPT NAME''' -
            &FILE=X21.FILE -
            &JCL=X21.CNTL(SUBSUB)
  *
  *
    RUN TASK (PGM=DMRTSUB -
            PARM= ("DSN=dsname, DISP=SHR", -
            "FILE      &FILE", -
            "JOB       &JOB", -
            "CLASS    &CLASS", -
            "DEST     &DEST", -
            "RPTNAME  &RPTNAME", -
            )) PNODE
  
```

Connect:Direct Exits and DMRTSUB

If your Connect:Direct system uses any of the standard exits (Security, RUN JOB, or RUN TASK), DMRTSUB functions with any or all of them without further modification. Both a RUN TASK and a RUN JOB exit are invoked by DMRTSUB, and any processing that takes place is done when using DMRTSUB.

Error Output

DMRTSUB writes any internal error information to the data set specified on the CDESTAE DD statement. Always include this DD statement in the Connect:Direct startup job stream.

Return Codes

The following table describes the DMRTSUB return codes.

Return Code	Meanings
04	A job stream record is truncated after substitution. Symbolic substitution is performed on columns 1–71 of the JCL record. Truncation is indicated when substitution causes the data in columns 1–71 to expand beyond column 71.
08	An error occurred parsing the parameters or the job stream source. A symbol length is greater than 19. A symbol name is absent in the parameter string.
12	No parameters are passed. An I/O error occurred. An input data set allocation parameter is not found. The symbol table overflowed. A file open error occurred. An input record length is greater than 256. A member locate error occurred.

Batch Compression Utility (CDSACOMP)

CDSACOMP is a batch utility that compresses and stores files in ZLIB-compressed format. You can transfer these compressed files using a Connect:Direct Process and store them on the remote node as compressed data. A user can execute CDSACOMP on the remote node to decompress the data and store it in the original format. These files cannot be decompressed using any other utility.

Note: You can also decompress files using a Connect:Direct Process. Refer to *Automatic Decompression* on page 191 for more information.

CDSACOMP supports physical sequential files (PS), KSDS, ESDS, and RRDS VSAM files. CDSACOMP cannot allocate VSAM files as DISP=(NEW,CATLG), so you must pre-define the VSAM output file when decompressing a file that was originally VSAM.

CDSACOMP does not support PDS (or PDSE) files as PDS, but you can use IEBCOPY to unload the PDS, then compress or decompress using CDSACOMP.

Parameters

You can pass the following parameters to CDSACOMP:

Parameter	Description
MODE= COMP DECOMP	Specify COMP to compress an input file. Specify DECOMP to decompress a compressed file.
REPORT=NORMAL DEBUG	Specify NORMAL to instruct CDSACOMP to produce a summary statistics report. This value is the default. Specify DEBUG to instruct CDSACOMP to produce a detailed statistics report.
ZLIB=(CMP= <u>1</u> n, WIN= <u>13</u> nn, MEM= <u>4</u> n)	<p>These three parameters change the settings of the extended compression utility. CMP indicates the compression level. Valid values are 1–9. The default value is 1. WIN indicates the window size. Valid values are 9–15. The default value is 13. MEM indicates the memory. Valid values are 1–9. The default value is 4.</p> <p>Caution: Changing these values can significantly impact performance and CPU overhead. Before you change the default values, see <i>Testing the Effects of Changing Values for Extended Compression Parameters</i> on page 190.</p>

Examples

The following sample JCL, found as member CDCMP in \$CD.SAMPLIB, compresses a physical sequential (PS) file and stores it as compressed data.

```
//CDCMP JOB (1004), 'BATCH COMPRESS', CLASS=N, MSGCLASS=X,
//          COND= (1, LT), REGION=0M
// *
// * ZLIB COMPRESS A FILE IN BATCH
// *
//COMP      EXEC   PGM=CDSACOMP, PARM= 'MODE=COMP'
//STEPLIB   DD     DISP=SHR, DSN=$CD.LINKLIB
//INPUT     DD     DISP=SHR, DSN=$CD.FILE1
//OUTPUT    DD     DSN=$CD.FILE1.COMP, DISP=(NEW, CATLG),
//          UNIT=SYSDA, SPACE=(TRK, (1500, 300), RLSE),
//          DCB=(BLKSIZE=27920, RECFM=U, DSORG=PS)
//SYSOUT    DD     SYSOUT=*, DCB=(BLKSIZE=133)
// *
```

The following sample JCL, found as member CDDECMP in \$CD.SAMPLIB, decompresses a compressed file and stores it in the original format.

```
//CDDECMP JOB (1004), 'BATCH COMPRESS', CLASS=N, MSGCLASS=X,
//          COND=(1,LT), REGION=0M
//*
//* ZLIB DE-COMPRESS A FILE IN BATCH
//*
//DECOMP   EXEC PGM=CDSACOMP, PARM='MODE=DECOMP'
//STEPLIB DD DISP=SHR, DSN=$CD.LINKLIB
//INPUT    DD DSN=$CD.FILE1.COMP, DISP=SHR
//OUTPUT   DD DSN=$CD.FILE1.NEW, DISP=(NEW,CATLG),
//          UNIT=SYSDA, SPACE=(TRK,(1500,300),RLSE),
//          DCB=(RECFM=FB, LRECL=80, BLKSIZE=27920)
//SYSOUT   DD SYSOUT=*
//*
```

Output

The following screen shows sample output from CDSACOMP.

```
SAC0000I CDSACOMP started, GMT =2001.275 19:39:25.00
SAC0001I OS Parm: MODE=COMP,
SAC0002I Options in effect: MODE=COMP  CMP= 1  WIN= 13  MEM= 4  REPORT=NORMAL
SAC0003I ZLIB Version=1.2.3
SAC0011I INPUT DSORG=PS RECFM=FB RECSZ= 80 DSN=CD.TESTFILE.M1
SAC0012I OUTPUT DSORG=PS RECFM=U RECSZ= 27,920 DSN=CD.TESTFILE.M1.COMP
SAC0021I Total Records Read = 12,500
SAC0022I Total Records Written = 5
SAC0023I Total Bytes Read = 1,000,000
SAC0024I Total Bytes Written = 124,584
SAC0031I Compression Percent = 88%
SAC0032I Compression Ratio = 8.0:1
SAC0033I TCB CPU Seconds = 2.627
SAC0034I Elapsed Seconds = 4.103
SAC0999I CDSACOMP utility completed successfully
```

You can also invoke CDSACOMP with the OUTPUT DD as DUMMY, indicating that the summary report is produced, but the output data is not stored. You must still provide the DCB attributes. The following screen shows an example.

```
//OUTPUT DD DUMMY,
//          DCB=(BLKSIZE=27920, RECFM=U, DSORG=PS)
```

Testing the Effects of Changing Values for Extended Compression Parameters

Before you change the default values of the extended compression parameters when using the CDSACOMP utility, review the benchmark figures showing different test results in the *Improving Performance* chapter in *Connect:Direct for z/OS Administration Guide*. You can also use this utility offline to determine the benefits of changing the default values in a test environment before actually using compression on live data in your production environment.

Although extended compression using ZLIB is available on a global basis using the ECZ initialization parameters and on a Process basis using the EXT parameters in the COPY statement, performing compression online consumes significant CPU resources. The CDSACOMP utility produces a report (see the sample report in *Output* on page 190), which shows how much the data was read, written, compressed, and how long it took to compress. Another report is produced when you use CDSACOMP to decompress data and restore it to its original format.

To use CDSACOMP to test compression, follow this procedure:

1. Prepare a file containing the typical type of data you transfer and the average amount of data involved in a transfer.
2. Transfer the file without using compression and record the time it takes for the transfer to complete.
3. To test the results of compressing and decompressing the file, run CDSACOMP three times on the test file using the following settings. After each compression run, run CDSACOMP to decompress the file. Keep the output reports for each test to compare results.
 - ◆ CMP=1,WIN=13,MEM=4 (the default settings)
 - ◆ CMP=1,WIN=1,MEM=14)
 - ◆ CMP=8,WIN=1,MEM=14)
4. To determine if it is worthwhile to use compression and what extended compression parameter values are most beneficial for your environment, examine the test results taking the following factors into consideration:
 - ◆ Amount of CPU and elapsed time it takes to compress and decompress the data
 - ◆ Amount of time it takes to send the data
 - ◆ Virtual memory space used to maintain the ZLIB internal control blocks
 - ◆ Virtual memory space used to allocate the compression window or history buffer
 - ◆ Type of data being sent, including its compressibility

Automatic Decompression

As an alternative to storing the data as compressed data on the remote node, you can decompress a file during Process execution. This functionality is available between z/OS and any other platform that supports ZLIB compression.

To perform automatic decompression, specify PRECOMP=YES in the SYSOPTS parameter in the FROM clause of the COPY statement. PRECOMP=YES indicates that the FROM data set is precompressed and instructs Connect:Direct to decompress the file as part of the Process. If you do not code SYSOPTS or set PRECOMP to NO, the file is sent in compressed format and the receiver must run CDSACOMP with MODE=DECOMP.

Note: If you are decompressing a file using CDSACOMP, you cannot allocate VSAM files as DISP=(NEW,CATLG). You must predefine the VSAM output file.

The following sample Process, found as member CDCOMPPS in \$CD.PROCESS, instructs Connect:Direct to decompress a file that is compressed using CDSACOMP.

Note: Refer to the \$CD.SAMPLIB data set for more sample Processes.

```

CDCOMPPS PROCESS SNODE=CD.OS390
STEP1 COPY FROM (DSN=$CD.FILE1.COMP -
              DISP=(SHR) -
              SYSOPTS='PRECOMP=YES' -
              PNODE) -
TO (DSN=$CD.FILE1.NEW -
   DISP=(RPL) -
   SNODE)

```

For more information on the Process language, see the [Connect:Direct Processes Web site](#).

Determine 'High-Water Mark' for a Period (DMRTVITL)

The program called DMRTVITL reports how many concurrent sessions (Processes) are running during the time period specified. Below is a sample RUN TASK that executes as a PNODE=SNODE Process when submitted. This Process must be force flushed to discontinue. The MCS console displays the number of sessions that ran during the specified period, and the time and date the high water mark was reached. These messages appear in the JESMSGLG segment of the Connect:Direct SYSOUT.

```

/*****/
/* RUN TASK FOR HIGH WATER PROCESS NUMBER; PNODE=SNODE */
/*****/
RTBB PROCESS SNODE=your.local.node
STEP1 RUN TASK (PGM=DMRTVITL, PARM=(C'60')) PNODE

```

The parameter, C, is the number of minutes desired in a measurement interval. The interval, 60, is an appropriate value to use. Since the Process that does the measuring runs as a PNODE=SNODE Process, the system adds a count of 2 to whatever the highest number of Processes that were running concurrently during a particular sample window. For example, if 6 other Processes were running simultaneously at a given time during the sample window, the value in the SVIT001I message will be '0008'. If no Processes were running during that sample time, the measuring Process is not counted, so the value in the SVIT001I message is '0000'.

Following is an example of the output of the DMRTVITL program:

```

SVIT001I SESSION HIGH WATER MARK OF: 0003 REACHED AT: 10:15:46 ON: 02000173

```

ADRSSU Interface Program (DMDSSIOX I/O Exit)

The DMDSSIOX I/O exit dynamically invokes the IBM ADRSSU utility to enable you to copy:

- ◆ SMS-COMPRESSED data without first uncompressing the data
- ◆ Wildcard-named files

Sample Processes

Three sample Processes are provided in the \$CD.PROCESS library distributed with Connect:Direct that you can use to prepare a Process to execute the DMDSSIOX I/O exit:

- ◆ DSSIOX1, which copies the SMS-COMPRESSED data set, KSTIC1.SMSTEST.DMDSSU.COMP.OUTLIST, from the PNODE to a data set of the same name on the SNODE but renames the SNODE data set to MWATL1.SMSTEST.DMDSSU3.** as part of the execution of the ADRSSU utility on the SNODE.
- ◆ DSSIOX2, which is similar to the DSSIOX1 process but uses symbolic substitution to name the input data set.
- ◆ DSSWILD, which is set up to copy wildcard-named files

Running Traces for the DMDSSIOX I/O Exit

The DMDSSIOX I/O exit automatically logs information to //ADRIOXLG each time the ADRSSU program is invoked regardless of the DEBUG bit settings. However, to log additional tracing information, use the setting, DEBUG=00008000 or DEBUG=04008000 in the DEBUG DDNames in the Connect:Direct startup JCL to generate an I/O Buffer Trace or an I/O Buffer Trace combined with a separate trace per task. DMDSSIOX I/O exit information logs to Rnnnnnnn while DSS exit information logs to Xnnnnnnn.

Limitations

Checkpoint/restart is not supported when using DMDSSIOX I/O exit for the following reasons:

- ◆ I/O exits in general including do not support Checkpoint/restart
- ◆ The IBM ADRSSU utility does not support Checkpoint/restart
- ◆ Control cards are allow for wild-carded input

In addition, the ADRSSU utility does not support DUMP/RESTORE of UNIX System Services user files such as PATH='/u/kstic1/abc.txt'. However, ADRSSU does support HFS files.

Connect:Direct FTP+ for z/OS (CDFTP)

Connect:Direct FTP+ for z/OS is an optional, licensed component of Connect:Direct for z/OS that provides a simple and dynamic migration path from unmanaged FTP sessions within z/OS job streams to a managed environment using Connect:Direct for z/OS with minimal change to the JCL.

Connect:Direct FTP+ for z/OS provides a simple, reliable, and secure way to transfer files between a Connect:Direct for z/OS installation at a central processing center and remote Connect:Direct sites. Connect:Direct FTP+ for z/OS runs as the client and supports Connect:Direct for z/OS, Connect:Direct for Windows, and Connect:Direct for UNIX as the remote system.

Components of Connect:Direct FTP+

The major components of Connect:Direct FTP+ are:

- ◆ **Connect:Direct FTP+ Manager**—Starts as a job stream or started task just like a Connect:Direct/Plex Manager. It handles Connect:Direct FTP+ Clients and is only a PNODE. It cannot be contacted by a remote Connect:Direct to start a process, and it cannot be contacted by a non-Connect:Direct FTP+ Client.

The Connect:Direct FTP+ Manager enables the TCP API and can be contacted by the ISPF IUI, Sterling Control Center, the Connect:Direct Browser User Interface, and the Connect:Direct JAI.
- ◆ **Connect:Direct FTP+ Client**—Starts a special case of a Connect:Direct Server. It builds and submits a Process which is scheduled by the Connect:Direct FTP+ Manager back to the Connect:Direct FTP+ Client using the server name for the PLEXCLASS. It is only a PNODE and cannot be contacted by a remote Connect:Direct to start a Process.
- ◆ **Connect:Direct FTP+ Plug-in**—Provides support for FTP commands that Connect:Direct does not support, such as CWD (change working directory), MKDIR (make a directory), RMDIR (remove directory), DIR (obtain a directory listing of files), and DELETE and MDELETE (deleting files).

When the JCL is changed to use Connect:Direct FTP+ for z/OS instead of FTP, all of the supported FTP commands are interpreted into appropriate COPY or RUNTASK and COPY steps and executed within the Connect:Direct FTP+ Client started in the user's address space.

Security Considerations

The Connect:Direct files, Secure+ parameter file and Netmap, are accessed through the Connect:Direct FTP+ Manager under its authority. The only Connect:Direct file that is accessed through the Connect:Direct FTP+ Client is the Connect:Direct Message File. All file accesses on the local system will be performed under the authority of the job userid (normal MVS operation). All file accesses on the remote system will be performed under the authority of the userid/password used for the SNODEID in the Connect:Direct FTP+ for z/OS Process.

To use the Connect:Direct FTP+ Client to connect to a remote Connect:Direct for z/OS system, the remote userid must have HFS file access on the remote Connect:Direct for z/OS system if the temporary file written by the plug-in on Connect:Direct for z/OS is an HFS file or if HFS files or directories are to be accessed.

Configure Connect:Direct FTP+ for z/OS

Before you can use Connect:Direct FTP+, complete the following tasks:

- ◆ *Install and Configure Connect:Direct FTP+ Plug-in* on page 195
- ◆ *Create CDFTP Procedure* on page 195
- ◆ *Create JCL to Start the Connect:Direct FTP+ Manager* on page 197
- ◆ *Update Netmap for Remote Nodes* on page 198
- ◆ *Start Connect:Direct FTP+ Manager* on page 199

Install and Configure Connect:Direct FTP+ Plug-in

The Connect:Direct FTP+ Plug-in must be installed and configured on the Connect:Direct nodes that Connect:Direct FTP+ for z/OS communicates with. The Connect:Direct FTP+ Plug-in is already included in the Connect:Direct for z/OS installation. If the remote node is a Connect:Direct for UNIX or Connect:Direct for Windows server, the Connect:Direct FTP+ Plug-in must be deployed on the remote system. For Connect:Direct FTP+ Plug-in installation and configuration instructions for Connect:Direct for Windows and Connect:Direct for UNIX, please refer to the latest *Connect:Direct FTP+ 1.0.08 Release Notes*.

Note: Make a note of the fully qualified installation directory of the plug-in (unless the remote node is a Connect:Direct for z/OS system) and the fully qualified name of the plug-in temporary file on the remote Connect:Direct server. You will need this information when you update the Netmap.

Create CDFTP Procedure

Build the CDFTP procedure and save it in your system's PROCLIB. The CDFTP procedure is executed in your modified FTP JCL.

The following is an example of a CDFTP procedure:

```
//CDFTP    PROC
//CDFTP    EXEC PGM=DMINIT,REGION=0M,
//    PARM='USER1.INIT.PARMLIB(GLOBINIT),CDFTP'
//STDENV DD DUMMY
//SYSTCPT DD DISP=(,DELETE),DSN=&&RES,
//    SPACE=(CYL,(1,1)),DCB=(DSORG=PS,RECFM=FB,
//    LRECL=80,BLKSIZE=24000),UNIT=SYSDA
//NETRC DD DISP=SHR,DSN=USER1.OS390.SRCLIB(NETRC)
//SYSTCPD DD DISP=SHR,DSN=SYS2.TCPIP.PARMS(TCPDATA2)
//STEPLIB DD DISP=SHR,DSN=USER1.CD5000.LOADLIB
//CDPLEX DD DISP=SHR,DSN=USER1.INIT.PARMLIB(SERVER)
//CDESTAE DD SYSOUT=*
//ESTAE DD SYSOUT=*
//NDMLOG DD SYSOUT=*
```

In this example, the Global initialization parameters file, **GLOBINIT**, and local initial initialization parameters file, **SERVER**, contain normal initialization parameters for starting a Connect:Direct/Plex Server. If **CDPLEX.SERVER=** is specified in the local initialization parameters, it is ignored and the JES jobname is used as the server name. The specification of

CDFTP as the first override parm after the specification of the location of the initialization parameters file identifies this DTF as a Connect:Direct FTP+ Client.

The SYSTCPT DD is required so that the Resolver will write its trace information into this file when a request to the Resolver, such as GETHOSTNAME, is issued. Connect:Direct FTP+ reads this trace information to get the resolved DATASETPREFIX from the Resolver to determine which FTP.DATA file to open and process.

A z/OS FTP client gets its configuration data from the FTP.DATA file and uses the following search order to find this file:

- ◆ DD card - //SYSFTP
- ◆ Tsoprefix.FTP.DATA
- ◆ Tsoid.FTP.DATA
- ◆ /etc/ftp.data
- ◆ SYS1.TCPPARMS(FTPDATA)
- ◆ tcpip_hlq.FTP.DATA

Connect:Direct FTP+ Client will attempt to find the correct FTP.DATA file using the following search order:

- ◆ DD card - //SYSFTP
- ◆ JobUserid.FTP.DATA
- ◆ /etc/ftp.data
- ◆ SYS1.TCPPARMS(FTPDATA)
- ◆ tcpip_hlq.FTP.DATA

Note: /etc/ftp.data must not be File Format NA or it will be ignored.

Note: tcpip_hlq is determined from DATASETPREFIX in the RESOLVER trace information written in the SYSTCPT DD. If DATASETPREFIX is not found, the default is "TCPIP".

If the FTP.DATA file is found, the following configuration parameters are honored by Connect:Direct FTP+ Client:

- ◆ PRIMARY—used for CDFTP.TEMPFILE allocation
- ◆ SECONDARY—used for CDFTP.TEMPFILE allocation
- ◆ SPACETYPE—used for CDFTP.TEMPFILE allocation
- ◆ KEYRING—used to override Secure+ keyring

Note: The FTP.DATA file is used by both z/OS FTP and Connect:Direct FTP+. Use ;CDFTPKEYRING if Connect:Direct FTP+ should use a different keyring than z/OS FTP.

- ◆ ;CDFTPKEYRING ;—used for a Connect:Direct FTP+ specific override of Secure+ Keyring

Note: Do not forget to put a semi-colon (;) in front of CDFTPKEYRING and a space followed by a semi-colon (;) at the end of CDFTPKEYRING, which makes this statement a comment to z/OS FTP, but will still be recognized by Connect:Direct FTP+ and processed.

;CDFTPKEYRING ;(or KEYRING if ;CDFTPKEYRING ; is not specified) will override the Secure+ specification of the keyring only if the Secure+ Parmfile was defined with a keyring. If the Secure+ Parmfile is defined to use a key database, the FTP.DATA file specification is ignored with a message to OUTPUT. To explicitly specify to use the Secure+ Parmfile definitions (keyring or key database, and to avoid the message to OUTPUT), code:
;CDFTPKEYRING ;

◆ **SECURE_DATACONN**—used as default for Secure+ transfers

The Connect:Direct FTP+ Client is a special case of a Connect:Direct/Plex Server in a Connect:Direct/Plex environment. The local initialization parameters file, **SERVER**, should specify:

```
CDPLEX.MANAGER=NO
```

For more information about global and local initialization parameters, please refer to the *Connect:Direct for z/OS Administration Guide*.

Create JCL to Start the Connect:Direct FTP+ Manager

Create JCL to start the Connect:Direct FTP+ Manager. Refer to the following JCL as an example:

```
//CD$MGR JOB (CDHOSTD), 'FTPUSER', PRTY=12, TIME=1440, CLASS=Q,
// REGION=0M, MSGLEVEL=(1,1), MSGCLASS=X, NOTIFY=&SYSUID
//*JOBPARM LINES=999999
//*****
//*
//*          CONNECT:DIRECT (Connect:Direct for z/OS MANAGER)          *
//*
//*****
// SET PGM=DMINIT, P=,
//     E='STAT.INIT=COLD, TCQ=COLD'
//NDMITST EXEC PGM=&PGM, REGION=0M,
// PARM='&P.USER1.INIT.PARMLIB(GLOBINIT), CDFTP, &E '
//CDPLEX DD DISP=SHR, DSN=USER1.INIT.PARMLIB(CDFTPMGR)
//CDSVCDMP DD DUMMY
//STEPLIB DD DSN=USER1.CD5000.LOADLIB, DISP=SHR
//DMPUBLIB DD DSN=USER1.PROCESS.LIB, DISP=SHR
//ESTAE DD SYSOUT=*
//CDESTAE DD SYSOUT=*
//NDMLOG DD SYSOUT=*
```

In this example, the JCL that starts the Connect:Direct FTP+ Manager is similar to JCL that starts the Connect:Direct/Plex Manager. The initparm override PARM **CDFTP** as the first override parm after the location of the global initialization parameters makes this a Connect:Direct FTP+ Manager.

The following local and global initialization parameters are forced with no message:

- ◆ TCP=OES
- ◆ SNA=NO
- ◆ CTCA=NO
- ◆ UDT=NO
- ◆ SECURITY=OFF (Connect:Direct FTP+ Client only. The Security parameter is allowed for the Connect:Direct FTP+ Manager)
- ◆ TCQ=COLD
- ◆ CONFIRM.COLD.START=NO
- ◆ TCP.FMH.TIMER=00:00:00
- ◆ TCP.RUNTASK.TIMER=00:00:00
- ◆ PROCESS.RETENTION=NO

If you specify any of the following Global Initialization Parameters, they are ignored:

- ◆ TCP.LISTEN
- ◆ UDT33.LISTEN
- ◆ MAXPROCESS
- ◆ MAXPRIMARY
- ◆ MAXSECONDARY

The local initialization parameters file for the Connect:Direct FTP+ Manager, **CDFTPMGR**, should specify:

CDPLEX.MANAGER=YES

For more information about global and local initialization parameters, please refer to the *Connect:Direct for z/OS Administration Guide*.

Update Netmap for Remote Nodes

Connect:Direct FTP+ for z/OS gets its configuration information about remote systems it connects to from the Connect:Direct Netmap.

Configure the following parameters in your Connect:Direct for z/OS Netmap for each remote system that Connect:Direct FTP+ for z/OS connects to:

- ◆ CDFTP.PLUGIN=<fully qualified installation directory of the Connect:Direct FTP+ Plug-in>
This parameter is not required when the Netmap entry defines a z/OS system.
- ◆ CDFTP.TEMPFILE=<fully qualified name of the Connect:Direct FTP+ temporary file>

The Connect:Direct FTP+ Plug-in creates a temporary file to store the results of a directory command RUNTASK operation.

Note: If no variables are used in the CDFTP.TEMPFILE, Connect:Direct FTP+ for z/OS appends the job userid and the ASID and CPUID to the CDFTP.TEMPFILE parameter as a further specification of uniqueness.

Several variables are available for substitution in the CDFTP.TEMPFILE specification. The following variables allow you to build an appropriate unique temporary file as a z/OS file:

- ◆ &userid;—The remote userid (lower case)
- ◆ &USERID;—The remote userid (upper case)
- ◆ &lusrid;—The local userid padded to eight characters with '\$'
- ◆ &lcpuid;—The first eight bytes of the local CPUID with "C" overlaying the first byte
- ◆ &asid;—The four character Address Space ID prefixed with "AS"

For example, you can specify **CDFTP.TEMPFILE=** with a variable ("**&userid;**"). Connect:Direct FTP+ for z/OS inserts the remote userid in place of the **&userid;**. In the following example, the userid specified in the script or NETRC file is **user1**.

```
CDFTP.TEMPFILE=/u/&userid;/cdftp/temp.file
```

CDFTP.TEMPFILE resolves to:

```
CDFTP.TEMPFILE=/u/user1/cdftp/temp.file
```

In the script or NETRC file, specify the userid in the case expected for the substitution.

Start Connect:Direct FTP+ Manager

Start the Connect:Direct FTP+ Manager as a job stream or started task the same way you start Connect:Direct/Plex Manager using the JCL created in the previous step.

The Connect:Direct FTP+ Manager must be running before the Connect:Direct FTP+ Client can successfully contact the Connect:Direct FTP+ Manager to perform its work.

For more information about starting a Connect:Direct/Plex Manager, please refer to the *Connect:Direct for z/OS Administration Guide*.

Modify FTP JCL

Use the following procedure to modify your FTP JCL to use Connect:Direct FTP+:

1. Change **EXEC PGM=FTP** to **EXEC PROC=CDFTP**.
2. Change the INPUT and OUTPUT DD statements and other step statements to specify the procedure step name. For example, change **//INPUT** to **//CDFTP.INPUT** and change **//OUTPUT** to **//CDFTP.OUTPUT**.

3. Change the first statement in the INPUT data stream to specify the remote Connect:Direct node name instead of the machine name for the FTP server.

The following is an example of JCL that executes the CDFTP procedure:

```
//CD$FTPJCL JOB (CDHOSTD), '-CDFTP+-', TIME=(1440), CLASS=Q,
// REGION=0M, MSGLEVEL=(1,1), MSGCLASS=X, NOTIFY=&SYSUID
/*JOBPARM LINES=999999
//LIBS JCLLIB ORDER=(USER1.OS390.SRCLIB)
//CDFTP EXEC PROC=CDFTP
//CDFTP.NETRC DD DISP=SHR, DSN=USER1.OS390.SRCLIB(NETRC)
//CDFTP.FTPTRACE DD SYSOUT=*
//CDFTP.OUTPUT DD SYSOUT=*
//CDFTP.INPUT DD *
MVSBB (exit
ebcdic
cd /u/user3
delimit
mkdir cdftptestlib
cd cdftptestlib
lcd 'USER1.CDFTP.TESTLIB'
delimit
bin
site sbdataconn=(IBM-037,ISO8859-1)
locsite sbdataconn=(IBM-037,ISO8859-1)
put CDFTPJCL
ebcdic
put CDFTPJCL CDFTPJC2
dir
ls
mdelete CDFTPJC*
cdup
rmdir cdftptestlib
close
quit
//
```

You must provide the following information in the JCL:

- ◆ Remote host—In this example, the hostname of the FTP server is changed to specify the Connect:Direct node name, **MVSBB**, that is in the Netmap. If the DNS name or IP address used by FTP is the nodename in the Netmap, no change is required. However, duplicating the entry in the Netmap and giving it the IP address or the DNS name as the node name may cause Netmap checking problems. In this case, the script should be changed to use the Netmap entry node name.

The remote host statement can also specify **(exit**, as shown in this example, or **(exit=nn** to specify that any error exits the script with a reason code (default is RC=8).

The remote host statement can also specify **(secure=off|ssl|tls** to specify security options.

If both **exit** and **secure=** are specified; only one open parenthesis can be used. For example, **MVSBB (exit 16 secure=off**.

- ◆ Remote userid and password—In this example, they are obtained from the NETRC file. If the remote userid and password are not specified in the NETRC file, they must be included in the FTP script after the remote host.

In the following sample excerpt of JCL that executes the CDFTP procedure, the remote host is **PROD.ZOS1**, the remote userid is **user1** and the password is **S1VX\$25B**.

```

...
//CDFTP.INPUT DD *
PROD.ZOS1
user1 S1VX$25B
cd /u/daily
...

```

You do not specify the use of Secure+ in the JCL. Secure+ is determined by the initialization parameters and the Secure+ parameter file definitions for the node named in the script.

To simplify matching up statistics in the remote node with Connect:Direct FTP+ for z/OS, the PROCESS name is the jobname and the process number is the jobid. The jobname is also used for the Connect:Direct FTP+ Client server name when contacting the Connect:Direct FTP+ Manager.

Note: Only the JCL interface is implemented. The ISPF interface (option 6;FTP) is not supported.

New PARMLIB and SAMPLIB Members

The Connect:Direct PARMLIB and SAMPLIB contain Initialization Parameters, JCL, and other samples that you can use and modify in your Connect:Direct FTP+ environment.

Member	Description	Location
CDFTPGBL	Global Initialization Parameters	PARMLIB
CDFTPMGR	Manager Local Parameters	PARMLIB
CDFTPSVR	Server (FTP+ Client) Local Parameters	PARMLIB
CDFTPJCL	CD FTP+ Manager Startup JCL	SAMPLIB
CDFTPCLI	CD FTP+ Client JCL	SAMPLIB
CDFTPPRC	CDFTP PROC	SAMPLIB
NETRC	Sample NETRC	SAMPLIB
CDFTPMAP	Sample NETMAP Entry	SAMPLIB

Supported FTP Commands

Connect:Direct FTP+ for z/OS separates the z/OS FTP commands into three categories:

- ◆ Commands completely supported
- ◆ Commands partially supported
- ◆ Commands accepted, but ignored.

All commands are case insensitive and can be abbreviated as specified in the z/OS FTP documentation.

Fully Supported FTP Commands

The following FTP commands are fully supported by Connect:Direct FTP+ for z/OS as described in the following:

Command	Description
APpend	Appends a data set on the local host to a file on the foreign host
BINary	Sets the transfer type to IMAGE
CD	Changes the working directory
CDUp	Changes to the parent of the current working directory
CLear	Sets the protection level for data transfers to CLEAR
CLose	Disconnects from the foreign host
COMpress	Sets the data transfer mode to compressed mode
CWd	Changes the working directory (Synonymous with CD)
DELEte	Deletes a single file on the foreign host
DELmit	Displays the delimiter character between the file_name and file_type
Dir	Lists the directory entries for files on the foreign host Note: Format is from Connect:Direct for z/OS Plug-in
EBcdic	Sets the transfer type to EBCDIC
Get	Copies a file from the foreign host to your local host Note: "(REPLACE" supported
GLob	Toggles globbing (the expansion of metacharacters in file names) for the MDelete, MGet, and MPut subcommands
LCd	Changes the working directory on the local host
LPwd	Displays the name of the active working directory on the local host
LS	Lists the names of files on the foreign host
MDelete	Deletes multiple files on the foreign host
MGet	Copies multiple files from the foreign host to the local host Note: "(REPLACE" supported
MPut	Copies multiple files on the local host to the foreign host
PRivate	Sets the protection level for data transfers to PRIVATE
PROtect CLEAR	PROTECT sets the protection level for data transfers on data connections. CLEAR sets SECURE=(ENC=N) in COPY steps.

Command	Description
PROtect PRIVATE	PROTECT sets the protection level for data transfers on data connections. PRIVATE sets SECURE=(ENC=Y) in COPY steps.
PUt	Copies a file on the local host to the foreign host
PWd	Displays the name of the active working directory on the foreign host
QUIt	Leaves the FTP command environment
REName	Renames a file on the foreign host
RMdir	Removes a directory
SENDSite	Enables or disables automatic transmission of the Site subcommand Note: Used to determine whether to propagate DCB or use SITE/LOCSITE

Partially Supported FTP Commands

The following FTP commands are supported by Connect:Direct FTP+ with limitations described in the Notes:

Command	Description	Note
AScii	Sets the transfer type to ASCII	Treated as EBCDIC
BIG5	Sets the transfer type to BIG5. BIG is the minimum abbreviation for BIG5.	Only to set DBCS= table in Connect:Direct
HAngeul	Sets the transfer type to HANGEUL	Only to set DBCS= table in Connect:Direct
JIS78kj	Sets the transfer type to JIS78KJ	Only to set DBCS= table in Connect:Direct
JIS83kj	Sets the transfer type to JIS83KJ	Only to set DBCS= table in Connect:Direct
Ksc5601	Sets the transfer type to KSC5601	Only to set DBCS= table in Connect:Direct
LMkdir	Create a directory on the local host	"(LIKE" ignored

Command	Description	Note
LOCStE	Specifies information that is used by the local host to provide service specific to that host system	Ignores all but: <ul style="list-style-type: none"> ◆ BLOCKS ◆ BLKSIZE ◆ BLOCKSIZE ◆ CONDISP ◆ CYLINDERS ◆ DIRECTORY ◆ LRECL ◆ PRIMARY ◆ RECFM ◆ SECONDARY ◆ TRACKS Errors out on: <ul style="list-style-type: none"> ◆ TRUNCATE ◆ WRAPRECORD
MKdir	Creates a directory on the foreign host	"(LIKE" ignored
MOde	Specifies the mode or data format of the transfer	MODE C honored as COMPRESS. All others ignored.
Open	Opens a connection to a foreign host	Port ignored
Pass	Supplies a password to the foreign host	No :userdata or account_information supported (ignored)
PROMpt	Toggles interactive prompting for MDelete, MGet, and MPut commands	Toggles prompt switch
PROXy	Executes an FTP subcommand on a secondary control connection	Treated as OPEN statement
PROTect SAFE	Sets SECURE=(ENC=Y) in COPY step	Treated as PROTECT PRIVATE
QUOte	Sends an uninterpreted string of data	Interprets command locally
SAfe	Sets the protection level on data transfers to safe	Treated as PRIVATE. Sets SECURE=(ENC=Y) in the COPY step.
SCHinese	Sets the transfer type to SCHINESE	Only to set DBCS= table in Connect:Direct

Command	Description	Note
Site	Sends information to the foreign host using site-specific commands	Ignores all but: <ul style="list-style-type: none"> ◆ BLOCKS ◆ BLKSIZE ◆ BLOCKSIZE ◆ CYLINDERS ◆ DIRECTORY ◆ FILETYPE JES ◆ LRECL ◆ PDSTYPE ◆ PRIMARY ◆ RECFM ◆ TRACKS Errors out on: <ul style="list-style-type: none"> ◆ TRUNCATE ◆ WRAPRECORD
SUnique	Toggles the storage methods	NAME/NONAME not supported Used to set DISP=NEW or DISP=RPL for output file
TChinese	Sets the transfer type to TCHINESE	Only to set DBCS= table in Connect:Direct
Type	Specifies the transfer type	Ignores all but: <ul style="list-style-type: none"> ◆ TYPE I = BINARY ◆ TYPE A = ASCII ◆ TYPE E = EBCDIC ◆ TYPE B 2 = EUCKANJI ◆ TYPE F = IBMKANJI ◆ TYPE B 5 = HANGEUL ◆ TYPE B 4 = JIS78KJ ◆ TYPE B 3 = JIS83KJ ◆ TYPE B 6 = KSC5601 ◆ TYPE B 1 = SJISKANJI ◆ TYPE B = SJISKANJI
User	Identifies you to a foreign host	No :userdata and account_information supported (ignored)

FTP Commands Accepted, But Ignored

The following commands are recognized, accepted, and ignored by Connect:Direct FTP+. These commands do not produce errors.

FTP commands that are not supported or recognized, are ignored with appropriate messages. However, if Connect:Direct FTP+ is in exit mode (**exit** or **exit=** specified in the remote host statement), these commands stop the script and terminate the step. This is how z/OS FTP behaves.

(LOC)SITE subcommands that are not supported or recognized are ignored and do not terminate the step, even in exit mode. The exceptions to this z/OS FTP behavior are the **(LOC)SITE WRAPRECORD** and **TRUNCATE** subcommands. If these subcommands are specified, they always terminate the script (**RC=8** when not in exit mode and with the user's return code when in exit mode).

- ◆ ?
- ◆ !
- ◆ ACCount
- ◆ BLock
- ◆ CCc
- ◆ CProtect
- ◆ DEBug
- ◆ DUMP
- ◆ EUCKANJI
- ◆ FEature
- ◆ File
- ◆ HElp
- ◆ Ibmkanji
- ◆ LANGuage
- ◆ LOCSTat
- ◆ NOop
- ◆ RECord
- ◆ REStart
- ◆ SENDPort
- ◆ SJiskanji
- ◆ SRestart
- ◆ STAtus
- ◆ STREam
- ◆ STRucture
- ◆ SYstem
- ◆ TSO
- ◆ UCs2
- ◆ Verbose

A

ABEND

A task that ends prematurely, or abnormally, due to an error that cannot be resolved by recovery facilities while the task is executing.

ACB

See *Access Method Control Block (ACB)*.

Access Method

A technique for moving data between main storage and input/output (I/O) devices.

Access Method Control Block (ACB)

A control block that links an application program to VSAM or VTAM.

Adapter

A hardware card that allows a device, such as a PC, to communicate with another device, such as a monitor, a printer, or other I/O device. See also Channel-to-Channel Adapter.

ADJACENT.NODE

An entry in the network map. Adjacent node entries define nodes in the network with which the local Connect:Direct may communicate. Each entry specifies a locally used Connect:Direct name, its associated network communications name, and session control parameters for these nodes.

API Pool

Identifies the APPLIDs to be used for API communication with the DTF.

Application Program Interface (API)

The Connect:Direct component that accepts commands from the Interactive User Interface (IUI), Batch Interface, the Operator Interface, or user-written program and places them in a format so that the user's request can be executed by the DTF. If there are errors, the API returns a message to the user. If there are no errors, the API sends the command to the DTF using a VTAM session.

APPLID

The name specified in the ACB macro that identifies the application program to VTAM. For Connect:Direct, these APPLIDs correspond to a DTF node name or an API APPLIDs.

Asynchronous Processes

Processes that occur without a regular or scheduled time relationship. Unexpected or unpredictable with respect to the instructions of the program or to time. Contrast with synchronous.

Attributes

Characteristics or properties that can be controlled, usually to obtain a required appearance; for example, the color of a line.

Authorization File

Connect:Direct file used to control access to Connect:Direct and identify commands that can be executed by user ID. This file can also be used in conjunction with security exit interfaces to support the secured point-of-entry feature.

B

Batch Interface

An interface where non-interactive programs are executed. The environment schedules their execution independently of their submitter. Connect:Direct users issue batch commands using DMBATCH, a Connect:Direct-supplied program.

Buffer

1. A portion of storage used to hold input or output data temporarily.
2. A routine or storage used to compensate for a difference in data rate or time of occurrence of events, when transferring data from one device to another.

Buffer Pool

A set of buffers that contains buffers of the same length.

C

Central Processing Unit (CPU)

The part of a computer that includes the circuits that control the interpretation and execution of instructions.

CF

See *Coupling Facility (CF)*.

Channel

1. A functional unit, controlled by a z/OS server that handles the transfer of data between processor storage and local peripheral equipment.
2. A path along which signals can be sent.
3. The portion of a storage medium that is accessible to a given reading or writing station.
4. In broadband transmission, a designation of a frequency band 6 MHz wide.

Channel-to-Channel (CTC)

Refers to the communication (transfer of data) between programs on opposite sides of a channel-to-channel adapter (CTCA). The CTCA for Connect:Direct for z/OS can be an ESCON CTC.

Channel-to-Channel Adapter (CTCA)

A hardware device that can be used to connect two channels on the same computing system or on different systems. The CTCA for Connect:Direct for z/OS can be an ESCON CTC.

Checkpoint/Restart

Eliminates the need to retransmit an entire file in the event of a transmission failure. A value in the COPY statement or CKPT initialization parameter specifies the checkpoint interval. If a copy procedure is interrupted, Connect:Direct will restart that copy at the last checkpoint.

CICS

See *Customer Information Control System (CICS)*.

Command Line Interface

Connect:Direct interface that allows users to submit Connect:Direct Processes and commands from their native command line environment.

Commands

An instruction that directs a control unit or device to perform an operation or a set of operations.

Connect:Direct users issue commands to initiate and monitor activity within the Connect:Direct system. Connect:Direct commands can be issued from the IUI, the operator console, a batch job, or a user application program.

Component

1. Hardware or software that is part of a functional unit.
2. A functional part of an operating system; for example, the scheduler or the Hold queue.

Compression

Storing data in a format that requires less space than usual. Data compression is particularly useful in communications because it enables devices to transmit the same amount of data in fewer bits. See also Variable Extended Compression.

Configuration

The arrangement of a computer system or network as defined by the nature, number, and main characteristics of its functional units. More specifically, the term configuration may refer to a hardware or software configuration. See also System Configuration.

Connect:Direct/Manager

The component of a Connect:Direct/Plex environment that handles the following functions:

- ◆ Interface connections
- ◆ Statistics file updates
- ◆ CKPT and TCQ/TCX file access
- ◆ TYPE file, AUTH file, NETMAP file, SECURE+ parameter file, and SECURE+ Digital Signature updates
- ◆ Workload balancing

Connect:Direct/Plex

A Connect:Direct system consisting of a Connect:Direct/Manager and one or more Connect:Direct/Servers in a system complex or parallel system complex.

Connect:Direct/Server

A Connect:Direct/Plex component that executes the Processes.

Connect:Direct/Stand-alone Server

A Connect:Direct system that is not part of a Connect:Direct/Plex.

Connectivity

A term used to describe the physical interconnections of multiple devices, computers, or networks employing similar or different technology and/or architecture together to accomplish effective communication between and among connected members. It involves data exchange and/or resource sharing.

Console

A logical device that is used for communication between the user and the system.

Coupling Facility (CF)

A special logical partition (LP) that provides high-speed caching, list processing, and locking functions in Parallel Sysplex.

CPU

See *Central Processing Unit (CPU)*.

Cross-System Coupling Facility (XCF)

A z/OS facility that allows multiple instances of the same application to communicate and share information with each other.

CTC

See *Channel-to-Channel (CTC)*.

CTCA

See *Channel-to-Channel Adapter (CTCA)*.

Customer Application

An application that does customer-specific processing.

Customer Information Control System (CICS)

An IBM-licensed program that enables transactions entered at remote terminals to be processed concurrently by user-written application programs. It includes facilities for building, using, and maintaining databases.

D

Daemon

A background task, process, or thread that intermittently awakens to perform some task or function and then returns to an idle state.

DASD

See *Direct Access Storage Device (DASD)*.

Database

1. A set of data, or a part or the whole of another set of data, that consists of at least one file and is sufficient for a given purpose or for a given data-processing system.
2. A collection of data fundamental to a system. See also Database Control (DBCTL), data entry database (DEDB), data sharing, and data sharing group.

Data Set

The major unit of data storage and retrieval, consisting of a collection of data in one of several prescribed arrangements and described by control information to which the system has access.

Data Sharing

In a Parallel Sysplex, the ability of concurrent subsystems or application programs to directly access and change the same data while maintaining data integrity.

Data Transmission Facility (DTF)

The nucleus component of Connect:Direct. The DTF controls information distribution to other Connect:Direct nodes in the network. Startup parameters that govern the overall activity of the DTF are defined within the initialization parameters.

In a Connect:Direct/Plex, the DTF consists of a Connect:Direct Manager and one or more Connect:Direct Servers.

Decipher

To convert enciphered data into clear data.

Decrypt

To convert encrypted data into clear data.

Default

Pertaining to an attribute, value, or option that is assumed when none is explicitly specified.

Direct Access Storage Device (DASD)

A physical device, such as an IBM 3390, in which data can be permanently stored and subsequently retrieved using licensed products like IMS and DB2, or using IBM supported access methods like VSAM in operating system environments like z/OS.

Directory

A list of files that are stored on a disk or diskette. A directory also contains information about the file, such as size and date of last change.

DTF

See *Data Transmission Facility (DTF)*.

Dynamic

Pertaining to an operation that occurs at the time it is needed rather than at a predetermined or fixed time.

E**Execution**

The process by which a computer carries out the instruction or instructions of a computer program.

Extended Submit Facility (ESF)

The facility that allows users to queue data transfer requests to a Connect:Direct node that is not active. This allows users to submit work to Connect:Direct, even if the Connect:Direct DTF is down.

F**File System**

The collection of files and file management structures on a physical or logical mass storage device such as a disk.

Format

1. A specified arrangement of things, such as characters, fields, and lines, usually used for displays, printouts, or files.
2. To arrange things such as characters, fields, and lines.

Function Management Header (FMH)

One or more headers, optionally present in the leading request units (RUs) of an RU chain, that allow one LU to (a) select a transaction program or device at the session partner and control the way in which the end-user data it sends is handled at the destination, (b) change the destination or the characteristics of the data during the session, and (c) transmit between session partners status or user information about the destination (for example, a program or device). Function management headers can be used with LU type 1, 4, and 6.2 protocols.

H

Hardware

The physical equipment as opposed to programs, procedures, rules, and associated documentation. Contrast with software.

Host (computer)

1. In a computer network, a computer that provides end users with services such as computation and databases and that usually performs network control functions.
2. The primary or controlling computer in a multiple-computer installation.

I

ICO

See *InterConnect Option (ICO)*.

Initialization

The preparation of a system, device, or program for operation. Connect:Direct initialization parameters specify alternate values for various parameters used during Connect:Direct start up.

Input/Output (I/O)

1. Pertaining to a device whose parts can perform an input process and an output process at the same time.
2. Pertaining to a functional unit or channel involved in an input process, output process, or both, concurrently or not, and to the data involved in such a process.

Interactive

Pertaining to a program or system that alternately accepts input and then responds. An interactive system is conversational; that is, a continuous dialog exists between user and system. Contrast with batch.

Interactive User Interface (IUI)

An ISPF screen and dialog component that allows users to define and submit Connect:Direct Processes as well as issue Connect:Direct commands that monitor and control administrative and operations activity. An IUI is also available for a CICS environment.

InterConnect Option (ICO)

A feature of Connect:Enterprise. This option provides an automatic, secure way to route application-produced distribution files from a Connect:Direct supported node to a Connect:Enterprise node for distribution, automatically distribute Connect:Enterprise batches to a Connect:Direct node upon arrival, and provide success or failure notification at each process step.

Interface

A shared boundary. An interface might be a hardware component to link two devices or it might be a portion of storage or registers accessed by two or more computer programs.

Interrupt

1. A suspension of a process, such as execution of a computer program caused by an external event, and performed in such a way that the process can be resumed.
2. In data communication, to take an action at a receiving station that causes the sending station to end a transmission.
3. To temporarily stop a process.

I/O

See *Input/Output (I/O)*.

I/O Service Units

A measure of individual data set I/O activity and JES spool reads and writes for all data sets associated with an address space.

IUI

See *Interactive User Interface (IUI)*.

J

Job Entry Subsystem (JES)

A system facility for spooling, job queuing, and managing job-related data.

L

LAN

See *Local Area Network (LAN)*.

Link

The combination of physical media, protocols, and programming that connects devices.

Load Module

A computer program in a form suitable for loading into storage for execution.

Local Area Network (LAN)

A data network located on the user's premises in which serial transmission is used for direct data communication among data stations. It services a facility without the use of common carrier facilities.

Local Cache

A buffer in local system storage that may contain copies of data entries in a CF cache structure.

LOCAL.NODE

An entry in the Network Map. The local node entry defines the logical Connect:Direct name of the local Connect:Direct DTF and its associated communications name. The local node entry also contains the name of the transmission queue and the SUPERUSR ID password, if specified.

Logical Connection

In a network, devices that can communicate or work with one another because they share the same protocol.

Logical Unit (LU)

In VTAM, the source and recipient of data transmissions. Data is transmitted from one logical unit (LU) to another LU. For example, a terminal can be an LU, or a CICS system can be an LU.

Logically Partitioned (LPAR) Mode

A CPC power-on reset mode that enables use of the PR/SM (Processor Resource/Systems Manager) feature and allows an operator to allocate CPC hardware resources (including CPUs, central storage, expanded storage, and channel paths) among logical partitions.

LU

See *Logical Unit (LU)*.

M

Main Storage

A logical entity that represents the program addressable portion of central storage. All user programs are executed in main storage.

Mainframe (z/OS server)

A large computer, in particular one to which other computers can be connected so that they can share facilities the z/OS server provides, for example, a z/OS computing system to which personal computers are attached so that they can upload and download programs and data.

Memory

The program-addressable storage from which instructions and other data can be loaded directly into registers for subsequent execution or processing.

Migration

Installing a new version or release of a program when an earlier version or release is already in place. See file migration.

Modal Statements

Statements (IF THEN, EIF, ELSE, EXIT, and GOTO) in Connect:Direct that allow you to alter the sequence of Connect:Direct Process execution based on completion of a previous Process step.

Module

A program unit that is discrete and identifiable with respect to compiling, combining with other units, and loading; for example, the input to or output from an assembler, compiler, linkage editor, or executive routine.

Multiprocessing

The simultaneous execution of two or more computer programs or sequences of instructions. See also Parallel Processing.

N

NCP

See *Network Control Program (NCP)*.

Netmap

See *Network Map*.

Network

A configuration of data processing devices and software connected for information interchange.

Network Control Program (NCP)

A program residing in a communication controller (for example, the IBM 3745 Communication Controller) that controls the operation of the communication controller.

Network Map

The VSAM file that identifies all valid Connect:Direct nodes and applids in the network. There is one Network Map (netmap) associated with each Connect:Direct node. There is one entry in that netmap for each of the other Connect:Direct nodes to which the local Connect:Direct node can initiate a session. The netmap entries also contain the rules or protocol to which the nodes will adhere when communicating.

Node

1. Any site in a network from which information distribution can be initiated.
2. In SNA, an endpoint of a link or junction common to two or more links in a network. Nodes can be distributed to Z/OS server processors, communication controllers, cluster controllers, or terminals. Nodes can vary in routing and other functional capabilities.

O

Offline

Not controlled directly by, or not communicating with, a computer. Contrast with online.

Online

Pertaining to equipment, devices, or data under the direct control of the processor. Contrast with offline.

Online Messages

The completion and error messages that are displayed online.

Operating System (OS)

The software that controls the execution of programs and that may provide services such as resource allocation, scheduling, input/output control, and data management. Although operating systems are predominantly software, partial hardware implementations are possible. Examples are z/OS, VSE/ESA, and VM/ESA.

Operator Interface

Allows Connect:Direct commands to be issued from the operator console. This interface also allows tailoring of Connect:Direct commands through a command list (CLIST) facility.

OS

See *Operating System (OS)*.

P

Parallel

1. Pertaining to a process in which all events occur within the same interval of time, each handled by a separate but similar functional unit; for example, the parallel transmission of the bits of a computer word along the lines of an internal bus.
2. Pertaining to the concurrent or simultaneous operation of two or more devices or to the concurrent performance of two or more activities in a single device.
3. Pertaining to the concurrent or simultaneous occurrence of two or more related activities in multiple devices or channels.
4. Pertaining to the simultaneity of two or more processes.

5. Pertaining to the simultaneous processing of the individual parts of a whole, such as the bits of a character and the characters of a word, using separate facilities for the various parts.

Parallel Processing

The simultaneous processing of units of work by many servers. The units of work can be either transactions or subdivisions of large units of work (batch).

Parallel Sessions

The capability of having two or more concurrently active sessions between the same set of two LUs. With parallel session support, Connect:Direct allows multiple, concurrent file transfers between two Connect:Direct nodes.

Parallel Sysplex

A sysplex with one or more coupling facilities.

Partitioned Data Set (PDS)

A data set in DASD storage that is divided into partitions, called members, each of which can contain a program, part of a program, or data.

Partitioned Data Set Extended (PDSE)

A data set organization that stores many different but related members. A PDSE contains an indexed directory and members that are similar to the directory and members of a partitioned data set (PDS) but offers architectural advantages that a PDS cannot. For example, space can be dynamically allocated and reclaimed using PDSEs, exploiting space more efficiently.

Primary Node (PNODE)

The Connect:Direct node on which the Process is being submitted. The primary node may also be referred to as the controlling or source node, but should not necessarily be interpreted as the sending node since PNODE can be the receiver. In every Process, there is one PNODE and one SNODE specified. The submitter of a Process is always the PNODE.

Process

A series of statements (which can be predefined and stored in a library) submitted through the API to initiate Connect:Direct activity, such as copying files, running jobs, and so on.

Process Statements

The statements that are used to build a Connect:Direct Process. They contain instructions for transferring files, running operating system jobs, executing programs, or submitting other Connect:Direct Processes. Process statements include COPY, RUN JOB, RUN TASK, SUBMIT, SYMBOL, and modals (conditional logic).

Processing Unit

The part of the system that does the processing, and contains processor storage.

Processor

A processing unit, capable of executing instructions when combined with main storage and channels.

Protocol

A specification of the format and relative timing of information exchanged between peer entities within a layer.

R**Record**

A set of data treated as a unit.

Recovery

To maintain or regain system operation after a failure occurs. Generally, to recover from a failure is to identify the failed hardware, to de-configure the failed hardware, and to continue or restart processing.

Remote Node

The Connect:Direct node that interacts with the local node during Process execution. The remote node is also referred to as the participating, receiving, target, destination, or secondary node (SNODE).

Resource Access Control Facility (RACF)

The facility that provides access control by identifying and verifying users to the system. RACF authorizes access to resources, logs unauthorized access attempts, and logs accesses to protected data sets.

Response Time

The amount of time it takes after a user presses the enter key at the terminal until the reply appears at the terminal.

Retry Interval

An installation parameter that specifies the interval, in minutes, that the retries mentioned in the Max Retries parameter will be performed.

Routing

The assignment of a path by which a transfer reaches its destination.

S

Secondary Node (SNODE)

The Connect:Direct node that interacts with the primary node (PNODE) during process execution. The secondary node (SNODE) can also be referred to as the participating, target, or destination node. Every Process has one PNODE and one SNODE.

Session

1. The entity through which a Connect:Direct PNODE initiates and executes one or more Processes to one or more SNODEs.
2. The entity through which a Connect:Direct SNODE receives one or more Processes.
3. In SNA, a logical connection between two network addressable units that can be activated, tailored to provide various protocols, and deactivated as requested.
4. The data transport connection resulting from a call or link between two devices.
5. The period of time during which a user of a node can communicate with an interactive system; usually it is the elapsed time between logon and logoff.
6. In network architecture, an association of facilities necessary for establishing, maintaining, and releasing connections for communication between stations.

Session Classes

The installation parameter that specifies the Process class groupings, priorities and number of Processes that can be concurrently executed on this Connect:Direct node.

Shared

Pertaining to the availability of a resource to more than one use at the same time.

SNA

See *Systems Network Architecture (SNA)*.

SNODE

See *Secondary Node (SNODE)*.

Standby Connect:Direct Manager

In an extended recovery environment, the backup Connect:Direct Manager that takes over work from the active Connect:Direct Manager when the active Connect:Direct Manager fails.

Standby Connect:Direct Server

In an extended recovery environment, the backup Connect:Direct Server that takes over work from the active Connect:Direct/Server when the active Connect:Direct Server fails.

Standby Connect:Direct System

In an extended recovery environment, the backup Connect:Direct system that takes over work from the active Connect:Direct system when the active system fails.

Statistics Facility

The Connect:Direct facility that records all Connect:Direct activities.

Statistics Files

A pair of VSAM data sets that hold Connect:Direct statistics records to document the history of a Connect:Direct Process.

Storage

A unit into which recorded data can be entered, in which it can be retained and processed, and from which it can be retrieved.

Subsystem

A secondary or subordinate system, or programming support, that is usually capable of operating independently of or asynchronously with a controlling system.

SYMBOL Statement

The Connect:Direct Process statement that allows you to build symbolic substitution values.

Symbolics

The parameters that allow one predefined Process to be used for multiple applications. For example, the file names for a COPY operation could be passed to the Process by the user submitting the Process.

Synchronous

1. Pertaining to two or more processes that depend on the occurrences of a specific event such as common timing signal.
2. Occurring with a regular or predictable timing relationship.

System

In data processing, a collection of people, machines, and methods organized to accomplish a set of specific functions.

System Configuration

A process that specifies the devices and programs that form a particular data processing system.

Systems Network Architecture (SNA)

A network architecture designed to provide compatibility among a wide variety of hardware and software products so that they can be used to build complex networks. It defines protocols, standards, and message formats to which different hardware and software products must conform.

The SNA network consists of network addressable units (NAUs), boundary function components, and the path control network.

T

TCQ

See *Transmission Control Queue (TCQ)*.

TCP/IP

See *Transmission Control Protocol/Internet Protocol (TCP/IP)*.

TDQ

See *Transient Data Queue (TDQ)*.

Terminal

A device that is capable of sending and receiving information over a link; it is usually equipped with a keyboard and some kind of display, such as a screen or a printer.

Throughput

1. A measure of the amount of work performed by a computer system over a given period of time, for example, number of jobs per day.
2. A measure of the amount of information transmitted over a network in a given period of time.

Transient Data Queue (TDQ)

A CICS temporary storage queue in which event data is stored so that a client application can retrieve the information.

Transmission Control Protocol/Internet Protocol (TCP/IP)

A set of public domain networking protocol standards that specify the details of how computers communicate, as well as a set of conventions for interconnecting networks and routing traffic.

Transmission Control Queue (TCQ)

A VSAM relative record data set (RRDS) used to hold all Processes that have been submitted to Connect:Direct.

U**UDT**

See *UDP (User Datagram Protocol) Data Transfer*.

UDP (User Datagram Protocol) Data Transfer

An alternative transport layer protocol to TCP designed for a high-latency, high-bandwidth connection to achieve higher data transfer rates than TCP over this type of connection.

V**Version 1/Version 2 Flows**

Version 1 and Version 2 Flows pertain to the internal Connect:Direct architecture and the way the Connect:Direct Function Management Header (FMH) protocol is performed between the

source and destination nodes. V1 flows were the original architectural design of Connect:Direct when it was used for VTAM SNA LU0 mainframe-to-mainframe data transfers. As the Connect:Direct product has matured and TCP/IP and LU6.2 protocols were adopted, continued usage of the original V1 FMH protocols was no longer practical.

The Version 2 FMH protocol introduced in the early 1990's changed the structure of the FMH protocol from DSECT-mapped to keyword (XDR) fields. In addition, Version 2 Buffer Headers and a new more efficient method of Checkpointing were included. The flexibility of the Version 2 architecture has enabled several other Connect:Direct product enhancements to occur, such as support for ZLIB compression, Secure+ Encryption, and CRC checking.

SNA LU0 and SNUF LU0 (for the OS/400) still use Version 1 FMH protocols, whereas TCP/IP, UDT, SNA LU6.2, and CTCA use Version 2 FMH protocols.

Virtual Storage (VS)

The storage space regarded as addressable main storage by the user of a computer system in which virtual addresses are mapped into real addresses. Virtual storage size is limited by the address scheme of the computer system and the amount of auxiliary storage available, rather than the actual number of main storage locations.

Virtual Storage Access Method (VSAM)

An access method for direct or sequential processing of fixed and variable-length records on direct access devices. The records in a VSAM data set or file can be organized in logical sequence by a key field (key sequence), in the physical sequence in which they are written on the data set or file (entry-sequence), or by relative-record number.

Virtual Telecommunication Access Method (VTAM)

The program that provides for workstation and network control. It is the basis of a System Network Architecture (SNA) network. It supports SNA and certain non-SNA terminals. VTAM supports the concurrent execution of multiple telecommunications applications and controls communication among devices in both single processor and multiple processor networks.

W

Wide Area Network (WAN)

A network that provides communication services to a geographic area larger than that served by a local area network.

X

XCF

See *Cross-System Coupling Facility (XCF)*.

A

Accounting data
 PNODE 63, 88
 SNODE 63, 90

ACTSESS transport state 136

ADMIN option 34

Administrative Options Menu 31

ADOPT V2 subtask state 135

ADDRSSU Utility 193

ALLOC ALLOCD transport state 136

ALLOC IMMED transport state 137

ALLOCATE task state 135

Allocation type 64

ANSWER V2 subtask state 135

API
 Connect:Direct 45
 signon 66
 writing 45

Application Program Interface (API) 9, 75

ARCHDSN parameter 155

ASMSAMP application 45

Asterisk 18

ATTACH task state 135

AUTH option 34

Automatic Decompression 191

B

Backslashes 22

Batch commands 25

Batch compression utility 188

Batch interface
 ddnames 27
 overview 25

 processing rules 26
 requirements 25
 return codes 28
 sample job 26
 sample job stream 26
 signon 61

Bracketing for special characters 22

Branch commands 34

Browsing table displays 24, 37

C

CALL V2 subtask state 135

Case in printed output 63

CASE parameter
 use 22
 with CHANGE PROCESS 112
 with DELETE PROCESS, FLUSH PROCESS, and
 SUSPEND PROCESS 117
 with SELECT PROCESS 123
 with SELECT STATISTICS 155
 with SIGNON 62
 with SUBMIT 85

Case sensitivity
 CHANGE PROCESS command 112
 DELETE PROCESS, FLUSH PROCESS, and
 SUSPEND PROCESS 117
 session level 62
 SUBMIT command 85

CASE SENSITIVITY parameter 97, 104

CAT SEARCH1 Allocate state 137

CCODE parameter 151

CDFTP 194

CDSACOMP 188

CF option 32, 94, 101

CHANGE PROCESS command
 batch interface 115
 description 110

Index

- examples 114
 - IUI 115
 - parameters 111
- Change Process screen 115
- Changing an existing Process 98
- CHECKPOINT parameter 103
- Checkpoint/restart
 - COPY parameters 142
 - examples 143
 - file 142
 - RUN TASK 144
- CKPT parameter 142
- CKPT.DAYS parameter 142
- CKPT.MODE parameter 141
- CKPTDSN parameter 142
- CLASS parameter 85, 112
- CLEANUP subtask state 135
- CLOSE transport state 136
- CLSDST transport state 136
- CNOS transport state 137
- COBSAMP application 45
- COMADDR parameter 62
- Command
 - example 18
 - label 17
 - parameters and subparameters 18
 - syntax 18
- Commas 19
- Comments 20
- Completion parameter 151
- Components 9
- Concatenation 21
- Condition parameter 151
- Connect:Direct exits and DMRTSUB 187
- Connect:Direct FTP+ for z/OS component 194
- Connect:Direct/Plex
 - configuration 11
 - description 10
 - messages 40
- Continuation marks 19
- Copy File Menu 102
- Copyfile - Receiving File screen 106
- CP option 33, 115
- Creating a Process with statement models 100

D

- DACTSESS transport state 136
- Data Transmission Facility 9
- DEBUG parameter 86
 - CHANGE PROCESS 112
- DEFINE transport state 137
- DELETE PROCESS command
 - batch 119
 - description 116
 - examples 118
 - IUI 119
 - parameters 117
- Delete session 69
- Deleting a nonexecuting Process 128
- Delimiters 21
- DELLOC transport state 137
- DESCRIPTION parameter 65
- DEST parameter
 - CHANGE PROCESS 112
 - SELECT PROCESS 122
- Detail statistics 164
- DF IUI option 98
- DF option 32, 94, 98
- DISPATCH task state 134
- Display output 24
- DISPLAY parameter 24
- Display statistics 165
- DISPLAY transport state 137
- Displaying signon defaults 59
- DMBATCH
 - ddnames 27
 - dreturn codes 28
 - EXEC parameters 27

DMCHLAPI
 DMBATCH return codes 28
 parameters 46
 return codes 48
 sample program 55
 with Connect:Direct API 45

DMMSGFIL ddname 27

DMNETMAP ddname 27

DMNOTFY2 179

DMNOTIFY 178

DMPRINT ddname 27

DMPUBLIB ddname 27

DMRTAMS 183

DMRTDYN 180

DMRTSUB 184, 187

DMRTWAIT 182

DP option 33

DSN parameter 85

DTF, see Data Transmission Facility

Dynamic allocation program 180

E

Elapsed time determination 149

End command 71

Errors 140

ESF parameter 63

Example 73

EXCLUDE parameter 151

EXEC status value 134

Execution queue
 commands 137
 status 127
 status values 134
 subtask state values 135
 task state values 134
 transport state values 136

EXTAREA parameter 48

Extended Recovery Facility 11

Extended Submit Feature field 63

EXTMSG parameter 48

EXTRC parameter 48

EXTSTRLN parameter 48

F

FILE I/O task state 134

FILE parameter
 description 24
 SELECT PROCESS 123
 SELECT STATISTICS 155

FLUSH PROCESS command
 batch 119
 description 116
 examples 118
 IUI 119
 parameters 117

FNAME parameter 155

FOLD parameter 63

Folding to upper case in printouts 63

FORCE FLUSH parameter 116, 117

FORCE SUSPEND parameter 116

FP option 33

Function keys 35

G

GENCB ACB transport state 136

GENCB EXTLST transport state 136

GENCB NIB transport state 136

GENCB RPL transport state 136

GET BUF subtask state 135

H

HANGUP subtask state 135

HC status value 138

HE status value 138

Help 24

HI status value 138

HO status value 138

Index

HOLD parameter 77, 87, 132
CHANGE PROCESS 112

Hold queue
commands 138
status values 138

HP status value 138

HR status value 138

HS status value 138

I

INACTIVE task state 134

INIT V2 subtask state 135

INITIAL COMMAND field 61

INQUIRE transport state 136

InterCONNECT Option 149

INVALKEY parameter 48

IOCTL V2 subtask state 135

IUI SUBMIT options
overview 94

L

Label 17

LOCK task state 134

Logical queues 131

M

M command 39

Master session 67, 72

MAXDELAY parameter 29, 88

Message Display screen 41

Message List 42

Message Maintenance screen 40

Message Summary screen 42

Messages

Connect:Direct 38
Connect:Direct/Plex 40
ISPF/PDF 38
printing 43

viewing by Message ID 40

MISC task state 134

Monitoring executing Processes 128

MOUNT task state 135

MS option 34, 68

MSG option 33

Multiple session SIGNON command
batch 68
IUI 68
usage 68

Multiple Session Signon screen 68

Multi-session signon 67

N

NDMCMDS ddname 27

NETEX I/O task state 134

NETMAP parameter 62

NETMAP.REFRESH parameter 112

Network Map

in DMBATCH 27

logon 62

retrieving records 169

SELECT NETMAP command 171

Select NETMAP or TCP Information screen 169

translating TCP/IP names with the batch interface 174

translating TCP/IP names with the IUI 173

NEWNAME parameter 88

NM option 33, 169, 173

NO REQUEST transport state 136

NO SESSION transport state 136

Node Name field 63

NODE parameter 63, 111

Non-master session 67, 73

NOTIFY parameter 88

O

Online Help 24

OPEN task state 135

OPEN transport state 136

- Operator console notify program 178
 - Operator Table
 - characteristics 37
 - commands 37
 - deleting a Process from 119
 - Execution queue status 127
 - suspending a Process from 119
 - viewing and controlling Processes 125
 - Operator Table - Select Process 125
 - Operator Table screen 126
 - Operator Table/Executing Queue screen 127
 - OPNDST transport state 136
 - OPNSEC transport state 136
 - Output display 24
- P**
- P=SNODE task state 134
 - PACCT parameter 63, 88
 - parentheses 19
 - PF keys 35
 - PLEXCLASS parameter 89, 113
 - PLISAMP application 45
 - PNAME parameter
 - CHANGE PROCESS 111
 - DELETE PROCESS 117
 - FLUSH PROCESS 117
 - SELECT PROCESS 122
 - SELECT STATISTICS command 152
 - SUSPEND PROCESS 117
 - PNODE Accounting Data field 63
 - PNODE parameter 89
 - PNODEID parameter 89
 - PNUMBER parameter
 - CHANGE PROCESS 112
 - DELETE PROCESS 117
 - FLUSH PROCESS 117
 - SELECT PROCESS 123
 - SELECT STATISTICS command 152
 - SUSPEND PROCESS 117
 - PR value 139
 - PR.CNTL status value 134
 - PRERECV transport state 137
 - Primary Options Menu 32
 - AUTH 36
 - CP 115
 - DF 98
 - DP 119
 - End 71
 - FP 119
 - MS 68
 - MSG 40
 - NM 169, 173
 - PS 119
 - SB 96
 - SD 60
 - SIGNOFF 71
 - SP 124
 - SS 159
 - SUS 119
 - SW 69
 - WHO 59
 - X 71
 - Primary space allocation 64
 - Print File Destination field 63
 - PRINT parameter
 - description 24
 - SELECT NETMAP command 173
 - SELECT PROCESS 123
 - SELECT STATISTICS 155
 - SELECT TCPXLAT command 175
 - SIGNON command 63
 - PROC parameter 85
 - Process control commands 109, 128
 - Process errors 140
 - Process execution
 - Example 79
 - Overview 75
 - Process management in a Connect:Direct/Plex 81
 - PROCESS NAME parameter 103
 - Process queuing 77
 - Process recovery 140
 - Process release from hold queue 110
 - Process routing routines 76
 - Process selection 77

Process step checkpoint 140
 PRTY parameter 89, 113
 PS option 33
 Putting nonexecuting Processes in the Hold Queue 128

Q

QUEUE parameter
 SELECT PROCESS 123
 Queues 76
 Queuing parameters 132
 Quotation marks 19

R

RA status value 138
 RCVFMH5 transport state 136
 RDY RCV subtask state 135
 RE value 139
 READ VTOC Allocate state 137
 RECEIVE transport state 136
 RECONNECT parameter 63
 RECV V2 subtask state 135
 Re-establish session 63
 REJECT transport state 136
 REJSESS transport state 136
 RELEASE parameter 113
 Releasing a Process from the hold queue 110, 128
 REQSESS transport state 136
 REQUEUE parameter 90
 RESTART parameter 113
 RETAIN parameter 76, 90, 132
 CHANGE PROCESS 114
 Return codes, DMCHLAPI 48, 53
 RH status value 138
 RSHUTD transport state 136
 RUN TASK
 checkpoint/restart 144
 completion codes 151

re-execution 145
 restart procedure 144
 user notification programs 177
 with DMNOTFY2 179
 with DMNOTIFY 178
 with DMRTAMS 184
 with DMRTDYN 180
 with DMRTSUB 184

RUNNING task state 134

S

SACCT parameter 63, 90
 SB option 32, 94
 Screens
 Change Process 115
 Copy File Menu 102
 Copyfile - Receiving File 106
 Message Display 41
 Message Maintenance 40
 Message Summary 42
 Multiple Session Signon 69
 Operator Table 126
 Operator Table/Executing Queue 127
 Primary Options Menu 32
 Select Netmap or TCP Information 169, 173
 Select Process 124, 128
 Select Statistics 159
 Selected Process 128
 Signon 59
 Signon Defaults 60
 Statistics Detail - Copy 164
 Statistics Display 165
 Statistics Summary 163
 Submit Process 96
 Swap/display user sessions 69
 User Authorization 36
 User Information 59
 SD option 33, 60
 Secondary space allocation 64
 Security at signon 57
 SELECT MESSAGE command 166
 SELECT NETMAP command
 batch 171
 IUI 169
 parameters 172

- Select Netmap or TCP Information screen 169, 173
- SELECT PROCESS command
 - accessing Operator Table 125
 - batch 124
 - description 121
 - example 124
 - IUI 124
 - parameters 121
- Select Process screen 124
- SELECT STATISTICS command
 - batch 156
 - examples 156
 - format 150
 - IUI 159
 - overview 150
 - parameters 151
 - sample output 158
- Select Statistics screen 159
- SELECT TCPXLAT command
 - batch 174
 - IUI 173
 - parameters 175
- Selected Process screen 128
- SEND CONFRMD transport state 137
- SEND ERROR transport state 137
- SEND RSP subtask state 135
- SEND SIG subtask state 135
- SEND transport state 136
- SEND V2 subtask state 135
- SERVER parameter 152
 - SELECT PROCESS 123
- SESSION EST transport state 136
- Session establishment retry 140
- SESSIONC transport state 136
- Sessions, maximum number 69
- SETLOGON transport state 136
- Sign off session 69
- SIGNOFF command
 - format 71
 - IUI 71
 - sequence 68
- SIGNON
 - batch 61
 - control 57
 - defaults 60
 - different nodes 72
 - IUI 58
 - multiple session, batch 68
 - multiple session, IUI 67
 - viewing parameters 59
- Signon Defaults screen 60
- SIMLOGON transport state 136
- SND RESPONSE transport state 136
- SNODE Accounting Data field 63
- SNODE parameter 90, 154
- SNODEID parameter 91
- SP option 33, 124
- SPACE parameter 64
- Special characters 21
- SPF option 34
- SS option 33, 159
- SS status value 134
- STARTT parameter
 - CHANGE PROCESS command 114
 - description 77
 - in Process queue 132
 - SELECT STATISTICS command 152
 - SUBMIT command 92
- Statement models 100
- Statistics Detail - Copy screen 164
- Statistics Display screen 165
- Statistics log 147
- Statistics Summary screen 163
- STATUS parameter
 - SELECT PROCESS 123
- Status value 133
- STEPLIB ddname 27
- STOPT parameter
 - SELECT STATISTICS command 153
- STOUTL utility 149
- SUBMIT command

batch 93
 description 84
 examples 93
 parameters 84
 Submit Process screen 96
 SUBMITTER parameter 112
 DELETE PROCESS 117
 FLUSH PROCESS 117
 SELECT PROCESS 123
 SUSPEND 117
 Submitting a Process
 using the IUI 94, 98, 101
 using the SUBMIT command 84
 Subparameters 18
 SUBTASK task state 134
 SUMMARY parameter 155
 Summary statistics 163
 SUSPEND PROCESS command
 batch 119
 description 116
 examples 118
 IUI 119
 parameters 117
 Suspending executing Processes 128
 SVC99 Allocate state 137
 SW option 34, 69
 SWAP NODE command 69
 Swap session 69
 Swap/Display User Sessions screen 69
 Switching nodes 68
 Symbolic resolution utility 184, 187
 Symbolic substitution
 description 19
 example 93
 Symbolic variables 187
 Synchronizing submitted processes 29
 SYSIN ddname 27
 SYSPRINT ddname 27

T

TABLE parameter
 description 24
 SELECT PROCESS 123
 SELECT STATISTICS 155
 Task state values 133
 TCA SCAN task state 135
 TCP Name/Address Translation Report screen 174
 TCQ, see Transmission Control Queue
 TCQ/TCX Repair Utility 140
 Temporary data set 64
 Terminating a master session 72
 Terminating a non-master session 73
 Terminating two sessions 72
 TIMEOUT parameter 65
 Timer queue
 commands 139
 status values 139
 TIMER task state 135
 TMPDD parameter 64
 TMPDSN parameter 64
 Transmission Control Queue (TCQ) 131
 TRANSPORT parameter 64
 Transport state values 133
 TYPE parameter 154

U

UICB fields 49
 UNIT parameter 64
 User Authorization screen 36
 User Information screen 59
 User interfaces 9
 User notification programs 177
 USER parameter 153
 USERID parameter 65
 Utility programs
 ADDRSSU 193

CDFTP 194
 CDSACOMP 188
 DMNOTFY2 179
 DMNOTIFY 178
 DMRTAMS 183
 DMRTDYN 180
 DMRTSUB 184, 187

X

X command 71
 XRF, see Extended Recovery Facility

V

validating Process syntax from IUI
 submit and validate 95
 without submitting the Process 95
 validating Processes
 options 95
 VOLSER parameter 65
 VSAM
 I/O task state 134
 VSAM AMS interface program 183
 VTAM
 automatic session retry 140
 I/O task state 134
 I/O transport state 136

W

WAIT FOR +DR transport state 136
 Wait queue
 commands 134
 status values 133
 WC status value 133, 134, 138
 WC value 139
 WHERE parameter 151, 166
 CHANGE PROCESS 111
 DELETE PROCESS 117
 FLUSH PROCESS 117
 SELECT NETMAP command 173
 SELECT PROCESS 122
 SELECT TCPXLAT command 175
 SUSPEND PROCESS 117
 WHO option 34, 59
 WT status value 133
 WX status value 133

