

CONNECT:Direct™ for VM/ESA

User's Guide

Version 3.2

CONNECT:Direct for VM/ESA User's Guide
Version 3.2
Edition Edition

This document was prepared to assist licensed users of the Sterling Commerce, Inc., CONNECT:Direct for VM/ESA system; its contents may not be used for any other purpose without prior written permission. The material contained herein is supplied without representation or warranty of any kind and is based on typical use. Any unusual use may produce unpredictable results. Sterling Commerce, therefore, assumes no responsibility and shall have no liability of any kind arising from the supply or use of this document or the material contained herein.

References in this manual to Sterling Commerce products, programs, or services do not imply that Sterling Commerce intends to make these available in all countries in which Sterling Commerce operates.

Restricted Rights: Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in FAR 52.227-19.

© 1998 Sterling Commerce, Inc.

All rights reserved, including the right to reproduce this document or any portion thereof in any form.

Printed in the United States of America.

CONNECT:Direct is a trademark of Sterling Commerce, Inc. All other brand or product names are trademarks or registered trademarks of their respective companies.



**STERLING
COMMERCE**

Product Enhancement Request Form

Company Name: _____

Address: _____

City: _____ **State:** _____ **Zip:** _____

Requester: _____ **Telephone:** _____

Product:

- | | |
|---------------------------------------------------------------|--------------------------------------------------------|
| <input type="checkbox"/> CONNECT:Direct for MVS | <input type="checkbox"/> CONNECT:Mailbox for VSE |
| <input type="checkbox"/> CONNECT:Direct for VSE/ESA | <input type="checkbox"/> CONNECT:Mailbox for MVS |
| <input type="checkbox"/> CONNECT:Direct for VM/ESA | <input type="checkbox"/> CONNECT:Mailbox for OS/400 |
| <input type="checkbox"/> CONNECT:Direct for MSP | <input type="checkbox"/> CONNECT:Mailbox SPC Option |
| <input type="checkbox"/> CONNECT:Direct for NetWare | <input type="checkbox"/> CONNECT:Mailbox for UNIX |
| <input type="checkbox"/> CONNECT:Direct for Tandem | <input type="checkbox"/> CONNECT:Supertracs for MVS |
| <input type="checkbox"/> CONNECT:Direct for UNIX | <input type="checkbox"/> CONNECT:Supertracs for VSE |
| <input type="checkbox"/> CONNECT:Direct for Stratus VOS | <input type="checkbox"/> CONNECT:Tracs for OS/2 |
| <input type="checkbox"/> CONNECT:Direct for OpenVME | <input type="checkbox"/> CONNECT:Tracs for MVS |
| <input type="checkbox"/> CONNECT:Direct for OpenVMS | <input type="checkbox"/> CONNECT:Tracs for MS-DOS |
| <input type="checkbox"/> CONNECT:Direct for OS/400 | <input type="checkbox"/> CONNECT:Tracs for VSE |
| <input type="checkbox"/> CONNECT:Direct for OS/2 | <input type="checkbox"/> CONNECT:Queue |
| <input type="checkbox"/> CONNECT:Direct for OS/2 Warp | <input type="checkbox"/> CONNECT:Firewall |
| <input type="checkbox"/> CONNECT:Direct for MS-DOS | <input type="checkbox"/> CONNECT:Remote for Windows NT |
| <input type="checkbox"/> CONNECT:Direct for Windows NT | |
| <input type="checkbox"/> CONNECT:Direct for Windows 95 | |
| <input type="checkbox"/> CONNECT:Direct for RemoteWare | |
| <input type="checkbox"/> CONNECT:Direct Requester for Motif | |
| <input type="checkbox"/> CONNECT:Direct Requester for Windows | |

Current Product Release Level: _____

Describe enhancement:

Over

How will enhancement be used?

Would you install a new release of this CONNECT product to get the enhancement?

Yes

No

Enhancement priority:

Need ASAP

Have immediate plan to use when available

Good idea; nice to have

Additional comments:

Mail or fax this completed form to:

**Sterling Commerce
Communications Software Group
Attention: Product Marketing
5215 North O'Connor Boulevard, Suite 1500
Irving, Texas 75039-3771
Fax: (972) 868-5142**

Thank you for your interest in the Sterling Commerce CONNECT Product Family.

Contents

Preface

What You Will Find in This Guide	xiii
How This Guide Is Structured	xiv
Chapter Overview	xiv
Notational Conventions	xv
Uppercase Letters	xvi
Uppercase and Lowercase Letters	xvi
Lowercase Letters	xvi
Bold Letters	xvi
Underlined Letters	xvi
Vertical Bars	xvi
Brackets	xvi
Additional Notations	xvii
Other CONNECT:Direct Publications	xvii

Chapter 1 About CONNECT:Direct for VM/ESA

CONNECT:Direct for VM/ESA Internal Components	1-1
User Interfaces	1-2
ISPF Interactive User Interface	1-2
Batch Interface	1-3
Operator Interface	1-3
Command Line Interface	1-3
Application Programs	1-4
Application Program Interface	1-4
Data Transmission Facility	1-4
CONNECT:Direct Concepts and Definitions	1-4
Processes	1-5
Commands	1-5
Transmission Control Queue	1-6

Sample Process Submission and Tracking	1-6
Building the Process	1-6
Issuing Commands to Track a Process	1-7
Submitting the Process to the TCQ	1-7
Monitoring the Progress of the Process	1-8
Determining the Outcome of the Process	1-8

Chapter 2 Introduction to the Command System

Using CONNECT:Direct Commands	2-1
Issuing the Commands	2-1
Supported Commands	2-2
Writing CONNECT:Direct Commands	2-3
Native Format	2-4
Example of a Command	2-5
Command Syntax	2-5
Asterisks	2-5
Commas	2-5
Continuation Marks	2-6
Parentheses	2-6
Symbolic Substitution	2-6
Single or Double Quotation Marks	2-7
Comments	2-7
Concatenation	2-8
Special Characters	2-8
Special Purpose Bracketing	2-9
Bracketing Backslashes	2-9
Special Considerations	2-9
Indicating Case Sensitivity	2-9
Session Case Sensitivity	2-10
Individual Command Sensitivity	2-10
Process Case Sensitivity	2-10
Indicating Selection Criteria	2-10
Indicating Output Destination	2-11

Chapter 3 Issuing Commands Through the Batch Interface

Batch Interface Requirements	3-1
Batch Interface Syntax	3-2
Batch Interface Processing Rules	3-2
Sample EXEC to Run the Batch Interface	3-3
DDNAMES for DMBATCH	3-3
Parameter Descriptions	3-4

	DMBATCH Return Codes with DMCHLAPI	3-4
	Synchronizing Submitted Processes	3-5
	Using the MAXDELAY Keyword Parameter	3-5
Chapter 4	Issuing Commands Through the Interactive User Interface	
	Primary Options Menu	4-1
	Understanding the Primary Options Menu	4-2
	Options to Branch to Submit Process Screens	4-2
	Viewing Statistics From the Primary Options Menu	4-2
	Options for Process Manipulation	4-3
	Viewing Your Environment	4-4
	Manipulating Sessions	4-5
	Branching to Other Menus	4-5
	Displaying Your User Authorization	4-6
	Obtaining Online Help	4-7
	Navigating Through The Help Screens	4-7
	Moving Through the Interactive User Interface	4-7
	Branching from the Command Line	4-7
	Branching to an Administrative Screen	4-8
	Branching to ISPF	4-8
	Using the PF Keys	4-8
	Branching from the IUI Summary Displays	4-9
	Browsing the IUI Displays	4-9
	Operator Tables	4-10
	Viewing Messages	4-11
	Viewing or Printing CONNECT:Direct Message Text	4-12
	Message Display Screen	4-13
	Message Summary Screen	4-13
	Message Print Screen	4-14
Chapter 5	Issuing Commands Through the Command Line Interface	
	CMS Command Line Interface File Definition	5-1
	Using the CMS Command Line Interface	5-2
	Using the Select Message Command Through the Command Line Interface	5-2
Chapter 6	Managing Sessions	
	Using the SIGNON Command	6-1
	SIGNON Command Format	6-2
	Required Parameter	6-2
	Optional Parameters	6-2

Using SIGNON Through the Batch Interface	6-4
Batch Interface SIGNON Example	6-4
Using SIGNON Through the IUI	6-5
Viewing Your Current Signon Parameters	6-6
Setting up Signon Defaults	6-6
Signing on to Multiple Sessions	6-7
Master Session	6-7
Non-Master Session	6-8
SIGNOFF Sequence	6-8
Switching Nodes	6-8
Using the Batch Interface for Multiple Session SIGNON ..	6-9
Using the IUI Multiple Session SIGNON Command	6-9
Instructions for Using Multiple Session SIGNON Through the IUI	6-10
Swapping Nodes	6-10
SWAP NODE Command Format	6-10
Using SWAP NODE Through the Batch Interface	6-11
Using SWAP NODE Through the IUI	6-11
Swapping Sessions	6-12
Deleting Sessions	6-12
Signing Off CONNECT:Direct for VM/ESA	6-13
SIGNOFF Command Format	6-13
Using SIGNOFF Through the Batch Interface	6-13
Using SIGNOFF Through the IUI	6-14
Sequencing the SIGNON, SWAP NODE, and SIGNOFF Commands	6-14
Example 1: Terminating Two Sessions Through One SIGNOFF	6-14
Example 2: Terminating a Master Session with SIGNOFF Command	6-15
Example 3: Terminating a Non-Master Session with the SIGNOFF Command	6-15

Chapter 7 Building, Modifying, and Submitting Processes

Submitting a Process Using the SUBMIT Command	7-2
SUBMIT Command Format	7-2
Required Parameters	7-2
Optional Parameters	7-3
Using SUBMIT Through the Batch Interface	7-11
Batch Interface SUBMIT Example	7-11
Using SUBMIT Through the IUI	7-12
Submitting a Predefined Process Using the SB IUI Option ..	7-13
Defining or Modifying a Process Using ISPF Editor (DF IUI Option)	7-14
How to Avoid Recovery Error Problems	7-15
Using Models to Define a Process	7-15

Changing an Existing Process	7-15
Creating a New Process	7-16
Submitting the Process	7-16
Using the DF Screen to Access Statement Models	7-16
Submitting a Process Using the SB IUI Option	7-17
Generating a COPY Process and Submitting it Using the CF IUI Option	7-17
Security Information	7-20
Sending and Receiving Environments	7-20
Copyfile - Sending File Screens by Environment	7-21
Copyfile - Receiving File Screens by Environment	7-22

Chapter 8 Controlling Processes in the TCQ

Controlling Processes Through Commands	8-1
Selection Criteria	8-1
Qualifying as Much as Required for Correct Match ...	8-2
Specifying a List	8-2
Filling in the Screens	8-2
Modifying a Process and Its Status in the TCQ	8-2
CHANGE PROCESS Command Format	8-3
Required Parameter	8-3
Optional Selection Subparameters	8-4
Optional Change Parameters	8-4
Batch Interface Use of CHANGE PROCESS	8-7
Batch Interface CHANGE PROCESS Example	8-7
Using CHANGE PROCESS Through the IUI	8-8
Suspending, Flushing, and Deleting Processes	8-9
SUSPEND PROCESS, FLUSH PROCESS, or DELETE PROCESS Command Format	8-11
Required Parameter	8-11
Selection Subparameters	8-11
Optional Parameter	8-12
Using the Commands Through the Batch Interface	8-12
SUSPEND PROCESS Examples	8-13
DELETE PROCESS Examples	8-13
FLUSH PROCESS Examples	8-13
Using the Commands Through the IUI	8-14
Sample of SUSPEND PROCESS Screen	8-14
Examining Processes in the TCQ	8-15
SELECT PROCESS Command Format	8-15
Required Parameter	8-16
Selection Subparameters	8-16
Optional Parameters	8-18
Using SELECT PROCESS Through the Batch Interface ..	8-18

Using SELECT PROCESS Through the IUI	8-18
Operator Table Summary of Selected Processes	8-20
Accessing Execution Queue Status on the Operator Table	8-21
Viewing and Controlling a Process Through the SELECTED PROCESS Screen	8-22

Chapter 9 Examining Process Results and Statistics

Statistics Log Records	9-1
Statistics Records Content	9-2
Elapsed Time Significance	9-3
Retrieving Statistics Records	9-3
SELECT STATISTICS Command Format	9-4
Required Parameter	9-4
Selection Subparameters	9-5
Optional Parameters	9-12
Using SELECT STATISTICS Through the Batch Interface	9-12
Examples	9-13
Sample Output	9-15
Using SELECT STATISTICS Through the IUI	9-16
Summarize Process Statistics	9-17
Display Statistics	9-18

Chapter 10 Viewing the Network Map

Retrieving Records from the Network Map File	10-1
SELECT NETMAP Command Format	10-1
Required Parameters	10-2
Optional Parameters	10-2
Batch Interface Use of SELECT NETMAP Command	10-2
Example	10-3
IUI Use of SELECT NETMAP Command	10-3

Chapter 11 Using the Shared File System

Understanding the Shared File System	11-1
Shared File System Components	11-1
Understanding How a Shared File System Request Works	11-2
Overview of a Shared File System Request	11-2
CONNECT:Direct DTF Commands for SFS File Interface Task	11-3
SFS Server Operator Commands	11-4

Chapter 12 Using the Utility Programs

User Notification Programs (DMNOTIFY and DMNOTFY2)	12-1
Operator Console Notify Program (DMNOTIFY)	12-2

User Notify Program (DMNOTFY2)	12-3
Dynamic Allocation Program (DMRTDYN)	12-4
Completion Code	12-5
Dynamic Allocation Parameters	12-5
Dynamic Allocation Guidelines	12-5
Description of the DMRTDYN Utility	12-6
Sample Program Uses	12-6
Sample Processes Using DMRTDYN	12-6
File Conversion Programs (DMRTPOMV and DMRTPOVM)	12-7
MVS-to-VM File Conversion Using DMRTPOMV	12-7
VM-to-MVS File Conversion Using DMRTPOVM	12-7
Punch a File Program (DMGPUN01)	12-8
Understanding the DMGPUN01 Program	12-8

Glossary

Index

Preface

This guide is for network operations staff members and programmers who use CONNECT:Direct commands to control Processes and environments and invoke utility programs.

This manual assumes you have a working knowledge of the IBM Virtual Machine/Enterprise System Architecture (VM/ESA) operating system and Interactive Systems Productivity Facility (ISPF). If you are not familiar with ISPF, refer to IBM's library of ISPF manuals.

What You Will Find in This Guide

This guide introduces you to the CONNECT:Direct command system.

The first two chapters provide:

- ▶ An overview of CONNECT:Direct
- ▶ Brief description of the CONNECT:Direct command format

The next three chapters describe three CONNECT:Direct interfaces:

- ▶ Batch Interface
- ▶ Interactive User interface (IUI)
- ▶ Command Line Interface (CLI)

The rest of the manual explains the commands and how to use these commands in the various interfaces.

Note: Examples are included to demonstrate the different tasks you can perform using these commands.

This guide covers the following topics:

- ▶ Understanding the command system
- ▶ Accessing CONNECT:Direct
- ▶ Building, modifying, and submitting Processes
- ▶ Controlling Processes
- ▶ Obtaining Process statistics
- ▶ Accessing environment commands
- ▶ Using utility programs

How This Guide Is Structured

The structure of this guide follows:

- ▶ The first four chapters of this guide:
 - Describes CONNECT:Direct for VM/ESA and explain its command system
 - Provides brief instructions on using the following interfaces:
 - Batch Interface (DMBATCH)
 - Interactive User Interfaces (IUI)
 - CMS Command Line Interface (CLI)
- ▶ In subsequent chapters you learn the following:
 - How to perform tasks using the commands
 - Required syntax
 - Valid parameters for each command

Note: Examples and sample screens are used to show you how to perform CONNECT:Direct tasks.

Chapter Overview

This manual consists of the following chapters:

- ▶ Chapter 1, *About CONNECT:Direct for VM/ESA*, provides an overview of CONNECT:Direct.
- ▶ Chapter 2, *Introduction to the Command System*, describes the CONNECT:Direct command system including tasks you can perform with the commands and how to write these commands.

- ▶ Chapter 3, *Issuing Commands Through the Batch Interface*, describes how to submit commands through the Batch Interface, DMBATCH.
- ▶ Chapter 4, *Issuing Commands Through the Interactive User Interface*, describes how to submit commands through the Interactive User Interface (IUI) and how to display or print the CONNECT:Direct message file.
- ▶ Chapter 5, *Issuing Commands Through the Command Line Interface*, describes how to submit commands through the Command Line Interface (CLI).
- ▶ Chapter 6, *Managing Sessions*, describes how to sign on to one or more sessions, set up signon defaults for your IUI, swap nodes, and sign off your sessions.
- ▶ Chapter 7, *Building, Modifying, and Submitting Processes*, describes how to submit a Process using the SUBMIT command in Batch Interface and the IUI. Using the IUI, you can edit and submit a predefined Process or generate and submit a COPY Process.
- ▶ Chapter 8, *Controlling Processes in the TCQ*, describes how to monitor the Processes and control their movement through the CONNECT:Direct Transmission Control Queue (TCQ) by selecting, suspending, changing, deleting, and flushing Processes.
- ▶ Chapter 9, *Examining Process Results and Statistics*, describes how to select and examine records from the CONNECT:Direct Statistics file.
- ▶ Chapter 10, *Viewing the Network Map*, describes how to view the CONNECT:Direct Network Map.
- ▶ Chapter 11, *Using the Shared File System*, describes the internal components of the Shared File System (SFS) and overviews how to request an SFS file.
- ▶ Chapter 12, *Using the Utility Programs*, describes how you use CONNECT:Direct utility programs including user notification, dynamic allocation, VSAM AMS interface, and file conversion programs and the symbolic resolution utility.

Glossary, defines terms used throughout this book.

Notational Conventions

The *CONNECT:Direct for VM/ESA User's Guide* uses certain notational conventions. This section describes the conventions used in this guide.

Uppercase Letters

Uppercase letters in the command format indicate that you type in information as shown.

Uppercase and Lowercase Letters

A statement, command, or parameter in uppercase letters followed by lowercase letters indicates an alternative to typing the entire command. For example, SElect PROCess means that you need only type SEL PROC for the command to be valid.

Lowercase Letters

Lowercase letters or words in commands or syntax boxes require substitution by the user. For example, PNODE=primary-node-name indicates that you must provide the name of the primary node.

Bold Letters

Bold print in syntax boxes indicates CONNECT:Direct commands and required parameters. For example, **DSN=filename** indicates that the parameter *DSN* is required.

Commands, Process statements, parameters, and special keys are sometimes bold in text to differentiate them from other words.

Underlined Letters

Underlining indicates default values for parameters and subparameters. For example, RETAIN=Yes | No | Initial specifies that the default for *RETAIN* is *NO*.

Vertical Bars

Vertical bars indicate that you can supply one of a series of values separated by the vertical bars. For example HOLD=Yes | No | Call specifies that *Yes* or *No* or *Call* is valid.

Brackets

Brackets indicate that information is optional. For example, STARTT=([date | day][,hh:mm:ssXM]) indicates that you can specify either a date or a day, a date or a day plus a time, or just a time.

Additional Notations

Code all commas and parentheses as they appear.

Process, as shown with a capital **P**, refers to a CONNECT:Direct Process.

Monospaced characters (characters of equal width) represent information for screens, commands, Processes, and reports.

Other CONNECT:Direct Publications

Use the *CONNECT:Direct for VM/ESA User's Guide* in conjunction with other CONNECT:Direct product documents. The following manuals make up the CONNECT:Direct for VM/ESA library:

- ▶ *CONNECT:Direct for VM/ESA Release Notes* is a document shipped with each CONNECT:Direct release that lists system requirements, maintenance updates, and enhancements.
- ▶ *CONNECT:Direct for VM/ESA Installation Guide* is a manual for planning and installing CONNECT:Direct for VM/ESA.
- ▶ *CONNECT:Direct for VM/ESA Administration Guide* is a manual for the administration and maintenance of CONNECT:Direct. This guide contains information on CONNECT:Direct tasks, native commands, diagnostics, and other administrative tasks.
- ▶ *CONNECT:Direct Process Guide* provides you with the information needed to write a CONNECT:Direct Process for the MVS, VSE/ESA, VM/ESA, MSP, OS/400, Tandem, UNIX, VOS, and VMS platforms.
- ▶ *CONNECT:Direct Console Operator's Guide* is a manual for the operator who initiates CONNECT:Direct for MVS, VSE/ESA, and VM/ESA data transfer activities from the operator console.
- ▶ *CONNECT:Direct Problem Isolation Guide* explains how to fix errors encountered when performing CONNECT:Direct for MVS, VSE/ESA, and VM/ESA functions.
- ▶ *CONNECT:Direct Quick Reference* is a capsulized reference of CONNECT:Direct for MVS, VSE/ESA, VM/ESA, and MSP Process statements, commands, and installation parameters.
- ▶ *CONNECT:Direct Technical Overview* is an introduction to the CONNECT:Direct product family and its data transfer applications.

- ▶ *CONNECT:Direct Event Services Support System Guide* provides information on the system architecture, system operation, and event data format for the Event Services Support.

About CONNECT:Direct for VM/ESA

CONNECT:Direct is a data transfer software product that distributes information and manages production activities among multiple mainframes, minicomputers, workstations, and personal computers in diverse operating system environments.

CONNECT:Direct allows data centers within and across networks to:

- ▶ Move large amounts of data
- ▶ Share information
- ▶ Schedule related application activities
- ▶ Automate data distribution
- ▶ Control and audit network activities
- ▶ Maintain network security
- ▶ Use a common command structure and environment-specific interfaces

CONNECT:Direct goes beyond traditional file transfer systems by eliminating the time-consuming, error-prone operator procedures associated with moving data. CONNECT:Direct capabilities extend from basic data movement functions to the management of data movement activities.

CONNECT:Direct for VM/ESA Internal Components

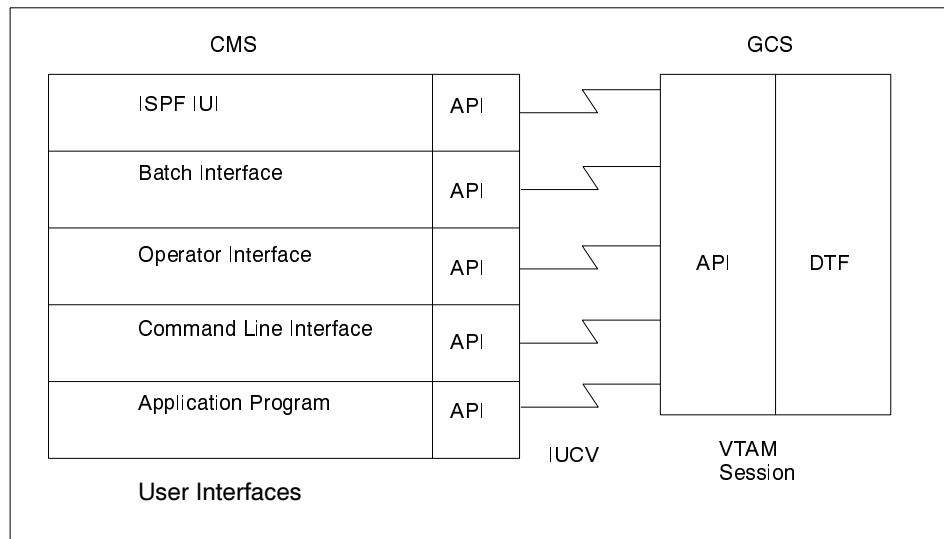
While CONNECT:Direct is running as an application on the VM/ESA operating system, the system components interact to execute the statements and commands submitted through the CONNECT:Direct interfaces.

The three internal components of CONNECT:Direct for VM/ESA are:

- ▶ Various user interfaces to CONNECT:Direct for VM/ESA
- ▶ Application Program Interface (API)
- ▶ Data Transmission Facility (DTF)

All statements and commands pass through the API regardless of the interface from which they are submitted.

The following figure shows a single CONNECT:Direct for VM/ESA system. Each piece of the diagram is discussed in the pages that follow.



User Interfaces

There are five ways to communicate with the CONNECT:Direct API:

- ▶ ISPF Interactive User Interface (IUI)
- ▶ Batch Interface
- ▶ Operator Interface
- ▶ Command Line Interface (CLI)
- ▶ Application Program

The following sections describe each interface.

ISPF Interactive User Interface

The IUI is a screen and dialog component running under the Interactive System Productivity Facility (ISPF) in CMS.

The IUI builds CONNECT:Direct commands based on information provided on the panels and submits them to the API. The IUI and API validate the syntax of statements as they are created, issuing messages

indicating acceptance or rejection of the request. CONNECT:Direct acts upon a request and then displays a message to the user.

Use the IUI to create and submit Processes, as well as to perform the following tasks (based on the security level of the user):

- ▶ Initiate file transfers
- ▶ Schedule file transfers
- ▶ Establish defaults for COPY attributes
- ▶ Monitor and modify CONNECT:Direct task activity
- ▶ Concurrently signon to other CONNECT:Direct sessions
- ▶ Display statistics online or off-line
- ▶ Display or print messages online
- ▶ Authorize new users
- ▶ Stop CONNECT:Direct
- ▶ Initiate traces
- ▶ Display, update, and insert Network Map entries
- ▶ Monitor and modify all CONNECT:Direct Process activity

See the *Issuing Commands Through the Interactive User Interface* chapter beginning on page 4-1 to learn how to use commands through the IUI.

Batch Interface

The Batch Interface allows you to issue any CONNECT:Direct command from a batch job stream to request CONNECT:Direct services. Refer to the *Issuing Commands Through the Batch Interface* chapter beginning on page 3-1 for more information on how to use the batch mode.

Operator Interface

The Operator Interface, which executes as a subtask of the DTF, enables the console operator to submit commands through CLISTs for system administration and maintenance. Refer to the *CONNECT:Direct Console Operator's Guide* for more information on the Operator Interface.

Command Line Interface

The Command Line Interface (CLI) allows requests of CONNECT:Direct services from CMS. The command line interface runs under CMS and uses the DMCHLAPI interface to communicate with the DTF running under GCS. Refer to the *Issuing Commands Through the Command Line Interface* beginning on page 5-1 for more information on how to use this interface to submit commands.

Application Programs

A high-level language Application Program Interface (API) allows user programs to interface directly to the CONNECT:Direct API. See the *Using the Program Interface* chapter in the *CONNECT:Direct for VM/ESA Administration Guide* for more information on the API.

Application Program Interface

The Application Program Interface (API) consists of the programs that allow the IUI, the Batch Interface, the Operator Interface, Command Line Interfaces, and application programs to communicate with the DTF or CONNECT:Direct Server. The API performs the following functions:

- ▶ Interprets commands from the various interfaces
- ▶ Validates the command format
- ▶ Passes the command across a VTAM session for DTF processing
- ▶ Receives the appropriate response or the requested data back from the DTF after the command is processed

Data Transmission Facility

The Data Transmission Facility (DTF) performs the following functions:

- ▶ Executes instructions, coded as Processes, passed to it from the API
- ▶ Starts and terminates all sessions
- ▶ Selects the next Process to execute
- ▶ Controls I/O requests for Shared File System (SFS) data sets
- ▶ Controls information distribution to other CONNECT:Direct nodes in the network

Note: Define start-up parameters that govern the overall activity of the DTF during installation.

CONNECT:Direct Concepts and Definitions

To understand the CONNECT:Direct tasks, you must understand these concepts:

- ▶ Processes
- ▶ Commands
- ▶ Transmission Control Queue

Refer to the *CONNECT:Direct for VM/ESA Installation Guide* and *CONNECT:Direct for VM/ESA Administration Guide* for more information on these concepts.

Processes

The *CONNECT:Direct* Process language consists of statements and parameters that provide instructions for copying files. You also use Processes for initiating such activities as running jobs and programs, and altering the sequence of Process step execution.

A Process consists of a Process definition statement (*PROCESS* statement) and one or more additional statements. Parameters further qualify Process instructions. A sample Process follows:

```
ONESTEP  PROCESS SNODE=SS.VM.USER01
STEP1    COPY FROM (SNODE DSN='USER01 DSN' ) -
                    LINK=(USERID,RPWORD,RR,191) -
                    CKPT=1K -
                    COMPRESS -
                    TO (DSN='USER01 OUT' -
                       LINK=(MYID,WPWORD,W,191))
```

You typically store Processes in your *CONNECT:Direct* Public Process Library but you can store them in your private Process library.

Refer to the *CONNECT:Direct Process Guide* for instructions on building Processes.

Commands

Use *CONNECT:Direct* commands to submit Processes to the *CONNECT:Direct* Transmission Control Queue (TCQ). You also use commands to manipulate Processes in the queue by flushing, deleting, selecting, or suspending them.

The following example submits the Process called COPY01 to the TCQ with a HOLD status of Yes.

```
SUBMIT PROC=COPY01 -
        HOLD=Y
```

Other commands allow you to sign on and off the *CONNECT:Direct* Data Transmission Facility (DTF), swap nodes, select and display statistics, or perform administrative functions such as maintain network maps, user authorities, and default types.

Transmission Control Queue

Use the Transmission Control Queue (TCQ) to control Process execution as CONNECT:Direct operates. CONNECT:Direct stores submitted Processes in the TCQ. The TCQ is divided into four logical queues:

- ▶ Wait
- ▶ Execution
- ▶ Hold
- ▶ Timer

CONNECT:Direct puts the Process in the appropriate queue based on Process statement parameters that affect scheduling. For example, the STARTT parameter indicates the day and time for a Process to execute.

CONNECT:Direct selects Processes in a first-in, first-out manner for execution within Process class and priority as sessions are available. You can access the queues and manipulate the Processes through the commands. For more information on the TCQ and how Processes progress through the queues, see the *Process Selection, Queuing, and Recovery* chapter of the *CONNECT:Direct for VM/ESA Administration Guide*.

Sample Process Submission and Tracking

This section shows a sample multi-step Process and how you track its submission, execution, and outcome.

Building the Process

Build the Process through the IUI, through ISPF, or some other method you select. The Process is stored as a member of your Process library. A typical multi-step Process follows:

```
COPY01  PROCESS  SNODE=CD.CHICAGO
STEP01  COPY FROM (DSN=&FROMDSN PNODE) -
        TO (DSN=&TODSN)
STEP02  IF (STEP01 >= 8) THEN
STEP03  RUN TASK (PGM=DMNOTIFY, -
        PARM=(' FAIL', &FROMDSN)) -
        PNODE
        EXIT
        ELSE
STEP04  RUN JOB (DSN='USER JOB', -
        LINK=(USER01, RDPASS, RR, 191), -
        BATCHID=VMBATCH -
        SNODE)
STEP05  COPY FROM (DSN=&TODSN SNODE) -
        TO (DSN=&FROMDSN)
EIF
```


A description of the previous Process follows:

COPY01 is a PROCESS statement that identifies the secondary or destination node (SNODE) as CD.CHICAGO.

STEP01 is a COPY statement that copies a file on the PNODE to a file on the SNODE. In this example, the &FROMDSN and &TODSN data set names are symbolic values that CONNECT:Direct resolves during submission of the Process. The ampersand (&) denotes symbolic names.

STEP02 uses Modal statements. If STEP01 fails (the return code is greater than or equal to 8), then STEP03 executes. If STEP01 completes successfully, then STEP04 and STEP05 execute.

STEP03 is a RUN TASK statement. The program DMNOTIFY executes within the CONNECT:Direct environment. The Process does not continue to the next step because the EXIT logic is executed; therefore, the Process ends.

STEP04 is a RUN JOB statement. The program 'USER JOB' is submitted to the operating system on the SNODE. The Process continues as soon as the program submission is successful; it does not wait for the program to execute.

STEP05 is another COPY statement. Once the file copies successfully, the Process terminates.

Issuing Commands to Track a Process

Use CONNECT:Direct commands to submit your Process to the CONNECT:Direct Transmission Control Queue (TCQ). Use commands to track your Process in the queue and to view its final result.

Submitting the Process to the TCQ

The following command submits the Process called COPY01 to the TCQ with the symbolic values &FROMDSN and &TODSN resolved.

```
SUBMIT PROC = COPY01 -  
          &FROMDSN='ABC FILEA', &TODSN='JKL FILEA'
```

If CONNECT:Direct encounters no errors in your SUBMIT command, it assigns a Process number to your Process, returns a *Submit Process Successful* message, and submits your Process to the TCQ.

Monitoring the Progress of the Process

After you submit the Process to the TCQ, use the `SELECT PROCESS` command to monitor the progress of the Process through the logical queues. An example of the `SELECT PROCESS` command follows.

```
SELECT PROCESS WHERE (PNAME=COPY01)
```

An example of `SELECT PROCESS` output follows on page 1-8.

In this example, you should note the *Queue* and *Status* fields. *Queue* indicates the logical queue where `CONNECT:Direct` has stored the submitted file. *Step Name* indicates what step the Process is in.

```
=====
                        SELECT PROCESS
=====

Prc Name   => COPY01                Queue    => EXEC
Prc Num    => 1                      Status   => EX
Subnode    => CD.DALLAS              Time     =>
Other Node => CD.DALLAS              Date     =>
Userid     => USERID                Day      =>
Prty       => 10                     Retain   => N
Class      => 1                      State    => P=SNODE
Step Name  => STEP1                  SubState => SEND
Function   => COPY                   Member   =>
Sent: Blks => 2,963                 Recs    => 2,963   RU's  => 43

-----

Prc Name   => COPY01                Queue    => EXEC
Prc Num    => 1                      Status   => -PR.CNTL
Subnode    => CD.DALLAS              Time     =>
Other Node => CD.DALLAS              Date     =>
Userid     => USERID                Day      =>
```

See page 8-18 for the various `SELECT PROCESS` display options available through the IUI, including summary and detail displays and operator tables.

Determining the Outcome of the Process

After the Process completes execution, use the `SELECT STATISTICS` command to review the statistics log for the Process. A sample command to generate a statistics log follows:

```
SELECT STATISTICS WHERE (PNAME=COPY01 EXCLUDE=(WTO))
```

A sample statistics log showing execution of a `CONNECT:Direct` Process follows. The log shows you the result of each step of the

Process, including the start and stop time and the number of records processed.

```

=====
CD.VM.NODE          SELECT STATISTICS      DATE : 04/06/1998
=====

Function    => PROCESS SUBMIT           Start Time => 12:13:40
Process Name => COPY01                   Stop Time  => 12:13:40
Process Num => 1                         Comp Code  => 00000000
Userid      => USERID                   Comp Msg   => SSPA001I
Secondary   => CD.VM.DALLAS              Step Name  =>
This Node   => CD.VM.DALLAS

-----
Function    => COPY                      Start Time => 12:13:41
Process Name => COPY01                   Stop Time  => 12:14:15
Process Num => 1                         Comp Code  => 00000000
Userid      => USERID                   Comp Msg   => SCPA000I
Primary     => CD.VM.DALLAS              Step Name  => STEP1
This Node   => CD.VM.DALLAS

From ( Pnode
  Dsn=DDR LISTING)
      recs => 6,723                       blks => 6,723
I/O Bytes => 861,243
  VTAM Bytes => 820,905
  Cmpr Perc => 4.7%
To ( Snode
  Dsn=!SPOOL USERID DDR4 LISTING)
      recs => 6,724                       blks => 0
I/O Bytes => 807,499
  VTAM Bytes => 820,905
  Cmpr Perc => - 1.6%

-----
Function    => COPY                      Start Time => 12:13:40
Process Name => COPY01                   Stop Time  => 12:14:15
Process Num => 1                         Comp Code  => 00000000
Userid      => USERID                   Comp Msg   => SCPA000I
Secondary   => CD.VM.DALLAS              Step Name  => STEP1
This Node   => CD.VM.DALLAS

From ( Pnode
  Dsn=DDR LISTING)
      recs => 6,723                       blks => 6,723
I/O Bytes => 861,243
  VTAM Bytes => 820,905
  Cmpr Perc => 4.7%
To ( Snode
  Dsn=!SPOOL USERID DDR4 LISTING)
      recs => 6,724                       blks => 0
I/O Bytes => 807,499
  VTAM Bytes => 820,905
  Cmpr Perc => - 1.6%

```

See page 9-1 for a description of the types of statistics records displayed in this report.

Introduction to the Command System

This chapter introduces you to the CONNECT:Direct command system and provides you with information about how to use CONNECT:Direct commands.

Using CONNECT:Direct Commands

Use CONNECT:Direct commands to submit and manipulate Processes in the TCQ. Process manipulation can include:

- ▶ Flushing
- ▶ Deleting
- ▶ Selecting
- ▶ Suspending
- ▶ Signing on and off the DTF
- ▶ Swapping nodes
- ▶ Selecting and displaying statistics

Other commands allow you to perform administrative functions such as maintain network maps, user authorities, and Type files.

Issuing the Commands

The methods of issuing commands that are described in this guide are:

- ▶ Batch Interface

The Batch Interface enables you to issue CONNECT:Direct commands in a batch environment. *The Issuing Commands Through*

the Batch Interface chapter beginning on page 3-1 describes how to use the Batch Interface.

► **Interactive User Interfaces**

The Interactive User Interface (IUI), an ISPF Dialog online interface, provides an easy-to-use method of command entry through fill-in-the-blank screens. *The Issuing Commands Through the Interactive User Interface* chapter beginning on page 4-1 describes how to use the IUI.

Note: Some command options which can be selected through the IUI do not have a batch equivalent. These exceptions are noted in the documentation where the option is described.

► **CMS Command Line Interface**

The CMS Command Line Interface (CLI) provides a method to communicate with the DTF running under the GCS. *The Issuing Commands Through the Command Line Interface* chapter beginning on page 5-1 describes how to use this interface.

See the *CONNECT:Direct Technical Overview* for other methods of issuing CONNECT:Direct commands.

Supported Commands

The following table lists the CONNECT:Direct commands, the IUI branching 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†	DT	Delete a record from the Type file
DELETE USER†	DU	Delete a user from the Authorization file
FLUSH PROCESS†	FP	Terminate an Executing Process
FLUSH TASK†	TF	Remove a task from the Execution queue
INQUIRE†	INQ	View information about CONNECT:Direct DTF status

(continued)

Command	IUI	Function
INSERT TYPE†	IT	Insert a record in the Type file
INSERT USER†	IU	Add a user to the Authorization file
MODIFY†	MD	Run CONNECT:Direct diagnostics
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†	TS	Display CONNECT:Direct system tasks status
SELECT TYPE†	ST	Examine the records in the Type file
SELECT USER†	SU	Examine user authorization
SIGNON		Connect to CONNECT:Direct DTF
SIGNOFF	X	Terminate a connection to the CONNECT:Direct DTF
STATISTICS†	STAT	Perform statistics functions
STOP CD†	SN	Stop CONNECT:Direct operation
SUBMIT	SB	Submit a Process for execution
SUSPEND PROCESS	SUS	Terminate Process execution
SWAP NODE	SW	Swap to another CONNECT:Direct node
UPDATE NETMAP†	UNM	Update the Network Map file
UPDATE TYPE†	IT	Update a record in the Type file
UPDATE USER†	IU	Change a user's privileges

† Administrative command. Refer to the *CONNECT:Direct for VM/ESA Administration Guide* for more information on administrative commands.

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.

The CONNECT:Direct IUI builds the commands in the proper format for you. If you are using the batch interface, operator interface, CMS

command line interface, or your own user-written application, you write the commands in the native format described in this section.

Native Format

This figure shows the structure of a CONNECT:Direct command in its native format.

```
Label Command Parameters and Subparameters
```

Label

(optional) allows you to identify CONNECT:Direct command with a label. A label must begin in position 1 and consists of a 1–8 character alphanumeric string, with the first character alphabetic.

Command

specifies the CONNECT:Direct function requested and must begin after position 1. Use one or more blanks or commas to separate the command from the parameters which follow it.

Parameters or Subparameters

specifies further instructions for the command. Separate your parameters by one or more blanks or commas. Parameters can be either keyword or positional.

Keyword parameters are usually followed by an equal sign and can have a set of subparameters. An example of a keyword parameter is *CASE* as follows.

```
SIGNON NETMAP=network.map.name CASE=Yes|No
```

Enter **positional parameters** in a specific order, with commas replacing any parameter not entered.

Note: 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.

In this command, *USERID* is an example of a keyword parameter, and *id,pswd,newpswd* are examples of positional subparameters.


```
SIGNON USERID=(id,pswd,newpswd)
```

A positional parameter or the variable information in a keyword parameter is sometimes a list of **subparameters**. These lists can include both positional and keyword parameters.

Example of a Command

The following CHANGE PROCESS command changes the 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 how to create a Process, see the *CONNECT:Direct Process Guide*.

Command Syntax

This section provides an overview of the syntax used to construct CONNECT:Direct commands and Processes. For a complete description of Process syntax and examples, see the *CONNECT:Direct Process Guide*.

Asterisks

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

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

```
USERID=ST*
```

Commas

Use commas to:

- ▶ separate items within a list
- ▶ 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

Parentheses enclose lists and associate a group of values. For example,

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

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 you resolve `&DATA` when you submit the Process.

```
DSN=&DATA
```

Single or Double Quotation Marks

The rules for using single and double quotation marks are as follows:

- ▶ Single-quoted strings allow the parsing of parameters as entered.
- ▶ Double-quoted strings allow the resolution of symbolic substitution in a quoted string.

The following example shows the use of single quotation marks to allow special characters or blanks to be embedded within a parameter or subparameter value.

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

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

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

Comments

Comments let you include descriptive information within a command.

Note: Comments are available for your convenience and do not affect CONNECT:Direct.

Follow these guidelines when you use comments within a command:

- ▶ Preceded by a slash-asterisk (/*) and followed by an asterisk-slash (*/).
- ▶ An asterisk (*) in position 1, followed by the comment. You must use this format when you submit a command to the Batch Interface.

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. The double bar must be preceded and followed by a blank. Use the double bar to join a long value that continues over multiple records.

For example, you can enter the following command.

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

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
}	closing brace

(continued)

Delimiters	Description
[opening bracket
]	closing bracket

Special Purpose Bracketing

Special characters are sometimes part of a string. To make this occur, enclose the string in bracketing characters such as backslashes (\), single quotation marks (' '), and double quotation marks (" ").

Bracketing Backslashes

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

This example uses bracketing backslashes in a command.

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

CONNECT:Direct resolves the command as follows.

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

Special Considerations

As you use the various commands, many designations are common to many commands. They include how you indicate case sensitivity, selection criteria, and output preference.

Indicating Case Sensitivity

The **CASE** parameter specifies whether parameters associated with accounting data, userid, 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 itself.

Session Case Sensitivity

The default for the session, as indicated by the **CASE** parameter of the SIGNON command, is NO. The session designation stays in effect for each command unless you override it at the command level.

Individual Command Sensitivity

You can change the case sensitivity at the individual command level to override the session case sensitivity as follows:

- ▶ In the IUI, the case sensitivity override stays in effect as long as you re-use the same panel. Case sensitivity reverts to the session default when you change panels.

The CASE parameter is usually indicated on the IUI by the following words.

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

Enter **Y** or **N** to this question.

- ▶ In native command mode, such as 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 Case Sensitivity

For commands that refer to Processes already in the TCQ, the case sensitivity parameter applies only 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 as well as the command.

Indicating Selection Criteria

Use the **WHERE** parameter to specify which records to select, change, or delete based on subparameters. Subparameters can be such values 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, on the CHANGE PROCESS screen you see this display.

```
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 **FILE** parameter causes CONNECT:Direct to pass back the output as unformatted records to the temporary file (NDMAP1 LISTING A). The format of the record is found in the sample macro library.

Once CONNECT:Direct puts the unformatted records in a temporary file, other programs can process the records and customize reports for specific needs. This is recommended for processing statistics records. You can archive a statistic record each day by means of a Generation Data Group (GDG). CONNECT:Direct provides sample job streams that extract statistics on a daily basis.

The **FILE** parameter specifies that the output is sent to a temporary file. Printed output is in a tabular format, the same as the **TABLE** parameter.

The **PRINT** parameter routes output of the command to the virtual printer. The name of the output is NDMMPI LISTING A.

The **TABLE** or **DISPLAY** parameter stores the output of the command in the CONNECT:Direct temporary file and displays it upon successful completion of the command. 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.

In the CONNECT:Direct IUI, the output parameter selection is usually found at the bottom of the screen as follows.

```
CMD: F ... FILE P ... PRINT D ... DISPLAY REPORT
```

The default value displays at the top right of the screen as follows.

```
CMD:DIS
```

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

Issuing Commands Through the Batch Interface

This chapter provides instructions on using the Batch Interface to issue CONNECT:Direct commands.

You request CONNECT:Direct services in the batch environment through the Batch Interface program, DMBATCH. 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.

Note: This manual contains information on user commands only. Refer to the *CONNECT:Direct for VM/ESA Administration Guide* for more information on administrative commands

Batch Interface Requirements

Your Batch Interface job must meet the following requirements:

- ▶ All Processes being submitted must have a filetype of NDMPROC.
- ▶ CONNECT:Direct commands used in the batch job stream must follow the syntax outlined in the *Writing CONNECT:Direct Commands* section beginning on page 2-3.
- ▶ Process statements cannot be used in the job stream. Use the SUBMIT command with a preexisting Process that contains Process statements.

Note: DMBATCH processing of the SUBMIT is asynchronous; the batch job will be notified if the SUBMIT was successful, but not whether the COPY itself was successful.

Batch Interface Syntax

Use these syntax rules when creating the SYSIN for the job stream.

- ▶ The DMBATCH input stream must be created in a file of fixed format and with an LRECL of 80 bytes.
- ▶ Commands begin after column 1. DMBATCH recognizes anything in column 1 of the input as a label; therefore, all commands must begin after column 1.
- ▶ Continuation marks (-) are used for any command requiring more than one line for completion. A continuation mark must be used on each line that continues. Place the continuation mark anywhere on the line as long as it is preceded and followed by a blank.
- ▶ Comments are indicated by an asterisk (*) in column 1.

Batch Interface Processing Rules

Follow these rules for entering 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 1,022 bytes.
- ▶ Each new CONNECT:Direct command in the job stream must start on a separate line. A command can be split 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.

The following example illustrates the batch interface processing rules.

```
SIGNON      USERID=(LYNN)      -  
            NODE=CD.BOSTON     -  
            GCSAPI=XYZ123  
SEL STAT WHERE (PNUM=24)  
SIGNOFF
```

Sample EXEC to Run the Batch Interface

This figure shows a sample EXEC to run the DMBATCH program.

```
/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */
/*
/* EXEC TO RUN CONNECT:Direct UTILITY DMBATCH                               */
/* - CHANGE MODE TO THE MODE WHERE THE MESSAGE FILE IS                     */
/*   LOCATED                                                                  */
/* - CHANGE HILQ TO THE HIGH LEVEL QUALIFIER                                */
/* - VERIFY THAT A "GLOBAL LOADLIB NDMV15"                                  */
/*   HAS BEEN DEFINED.                                                       */
/*
ARG      BATNAME  BATTYPE  BATMODE
        'DLBL IJSYSCT V DSN MASTCAT (PERM'
        'DLBL DMMSGFL V DSN HLVSAM.MSG ( VSAM'
FILEDEF DMPRINT T
FILEDEF SYSIN DISK BATNAME BATTYPE BATMODE
        'OSRUN DMBATCH PARM=YYSLYNN'
```

To run this BATCH EXEC, enter the following:

```
BATCH  BATNAME  BATTYPE  BATMODE
```

In this example, BATNAME BATTYPE BATMODE is the file where the SYSIN data exists.

DDNAMES for DMBATCH

The following are descriptions of required ddnames for the DMBATCH program as shown in the sample EXEC on page 3-3.

The **DMMSGFL** statement indicates the CONNECT:Direct message file that contains CONNECT:Direct messages.

The **DMPRINT** statement indicates the destination of the job output from DMBATCH used to collect output messages such as error messages.

Note: Always check DMPRINT output for verification of the processing that has just occurred.

The **SYSIN** statement indicates the location of the CONNECT:Direct commands. It can be represented as a sequential file, MACLIB or TXTLIB member.

Parameter Descriptions

The characters in the PARM keyword on the DMBATCH EXEC statement are required parameters. Their values are shown in the following table.

They 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. There are no default values. The output is displayed to the data set defined by DMPRINT.

Position	Value	Meaning
1	Y	Display the command string that was executed
	N	Do not display the command string that was 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 there is a return code of zero from the API
	L	Display long message text when there is a return code of zero from the API
	N	Do not display message text when there is a return code of zero from the API
4	S	Display short message text when there is a non-zero return code from the API
	L	Display long message text when there is a non-zero return code from the API
	N	Do not display message text when there is a return code greater than zero from the API
5	Y	Display the data that was generated in the temporary file
	N	Do not display the data that was generated in the temporary file
6	N	Reserved
7	N	Reserved

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.

If DMCHLAPI detects an error, CONNECT:Direct issues special return codes based on the situation. If you receive a DMBATCH return code greater than 8, check the following list for error information.

Return Code	Meaning
000000C	A session was lost in a multiple session environment.
0000010	The master session was lost.
0000014	The master session was signed off successfully.
0000018	The master session signon failed.
000001C	A non-master signon failed.
0000020	The high-level interface program, DMCHLAPI, received an invalid number of input parameters.
0000024	The output specification included an invalid parameter.
0000028	Invalid pointer to the UICB was passed to DMCHLAPI.

Synchronizing Submitted Processes

You can synchronize the operation of batch job streams with the MAXDELAY keyword parameter on the SUBMIT or PROCESS statement. The following information describes how to use each of these methods.

Using the MAXDELAY Keyword Parameter

The MAXDELAY keyword allows the suspension of execution of the DMBATCH jobstep until the submitted process either completes or a specified interval of time elapses. For a detailed description, refer to the definition of the MAXDELAY parameter on page 7-4 of this manual.

The following example shows the MAXDELAY keyword coded on the SUBMIT statement.

```
SIGNON USERID=(USER01,PASSWRD)
SUBMIT PROC=PROCAA MAXDELAY=UNLIMITED
SIGNOFF
```

The MAXDELAY keyword can be placed on the PROCESS statement for PROCAAA as shown in the next 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)
```

Issuing Commands Through the Interactive User Interface

This chapter provides general instructions for using the Interactive User Interface (IUI) to issue CONNECT:Direct commands. You issue CONNECT:Direct IUI commands through two menus: Primary Options Menu and Administrative Options Menu. The fill-in-the-blank screens of the IUI help guide you through many of the Processes you use.

Primary Options Menu

The Primary Options Menu displays CONNECT:Direct functions that 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:mm.dd.yyyy *
  SS - VIEW STATISTICS FOR A COMPLETED PROCESS      * TIME:hh:mm      *
                                                    *
  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
MS - SIGN ON TO MULTIPLE CONNECT:Direct NODES CONCURRENTLY
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
ADMIN - PERFORM CONNECT:Direct ADMINISTRATIVE FUNCTIONS

```

Understanding the Primary Options Menu

The Primary Options Menu displays available CONNECT:Direct functions that you can branch to perform a specified task. To request one of the functions, type the option identification for the function on the command line (**CMD==>**) and press **ENTER**.

For example, type **CF** on the command line and press **ENTER** to branch to the Copy a File screen. A brief explanation of each command follows.

Options to Branch to Submit Process Screens

The following options allow you to branch to screens that help you create and submit Processes to the TCQ:

- ▶ **CF** Copy a File
- ▶ **SB** Submit a Predefined Process
- ▶ **DF** Define a Process Using ISPF Edit

A definition for each option follows.

CF

displays the main Copy File Menu where you can set up a COPY Process. Refer to the menu descriptions beginning on page 7-19 for a detailed summary of this menu.

SB

displays the Submit Process screen where you may submit a predefined Process. Refer to the *Submitting a Predefined Process Using the SB IUI Option* section beginning on page 7-13 for more information on this menu.

DF

displays the Process Definition Screen where you define or update a Process using the ISPF edit function and then submit the Process. Refer to the *Defining or Modifying a Process Using the ISPF Editor (DF IUI Option)* section beginning on page 7-14 for more information.

Viewing Statistics From the Primary Options Menu

Use the **SS** option to view statistics related to your completed Processes.

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. Refer to the *Using SELECT*

STATISTICS Through the IUI section beginning on page 9-16 for more information on this menu.

Options for Process Manipulation

Use the following options to branch to screens where you can manipulate Processes in the TCQ:

- ▶ **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

A definition for each option follows.

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. Refer to the *Using CHANGE PROCESS Through the IUI* section beginning on page 8-8 for more information about this menu.

DP

displays the Delete Process screen where you can remove a nonexecuting Process from the TCQ. Refer to the *Using the Commands Through the IUI* section beginning on page 8-14 for more information on this menu.

FP

displays the Flush Process screen where you can remove an executing Process from the TCQ. Refer to the *Using the Commands Through the IUI* section beginning on page 8-14 for more information on this menu.

SP

displays the Select Process screen where you can view the status of Processes that you submitted (placed in the TCQ) for execution. Refer to the *Using SELECT PROCESS Through the IUI* section beginning on page 8-18 for more information on this menu.

PS

displays the Suspend Process screen where you can put an executing Process in the Hold TCQ. Refer to the *Using the Commands Through the IUI* section beginning on page 8-14 for more information on this menu.

Viewing Your Environment

Use the following options to branch to screens to view your CONNECT:Direct environment:

- ▶ **SD** View/Change Your CONNECT:Direct Sessions
- ▶ **WHO** View Characteristics of IUI Environment
- ▶ **AUTH** View Your FUNCTION AUTHORIZATION
- ▶ **NM** View Information in Your NETWORK MAP
- ▶ **MSG** View Your CONNECT:Direct MESSAGE TEXT

A definition for each option follows.

SD

displays the Signon Defaults screen where you can examine and change your signon default values. See the *Setting Up Signon Defaults* section beginning on page 6-6 for more information on this menu.

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. Refer to the *Viewing Your Current Signon Parameters* section beginning on page 6-6 for more information on this menu.

AUTH

displays the User Authorization screen. Refer to the *Displaying Your User Authorization* section beginning on page 4-6 for information on this display.

NM

displays the Select Network Map screen, where you choose to:

- ▶ display or print the defined nodes from the Network Map file
- ▶ translate TCP/IP host names and network addresses

See page 10-3 for more information about this option.

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. Refer to the *Viewing or Printing CONNECT:Direct Message Text* section beginning on page 4-12 for more information on this menu.

Manipulating Sessions

Use the following options to sign on and swap among multiple sessions:

- ▶ **MS**
- ▶ **SW**

A definition for each option follows.

MS

displays the Multiple Session Signon screen where you can sign on to another DTF session. Refer to the *Using the IUI Multiple Session SIGNON Command* section beginning on page 6-9 for more information on this option.

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. Refer to the *Using SWAP NODE Through the IUI* section beginning on page 6-11 for more information on this menu.

Branching to Other Menus

The following options allow you to branch to other menus:

- ▶ **SPF**
- ▶ **ADMIN**

A definition for each option follows.

SPF

branches to an ISPF session, a level beneath the CONNECT:Direct panels. Refer to the *Branching to ISPF* section beginning on page 4-8 for more information on this option.

ADMIN

displays the CONNECT:Direct Administrative Options Menu. This menu displays a series of command options used for administering the system. See the *CONNECT:Direct for VM/ESA Administration Guide* for the menu produced by this option.

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. The AUTH command is available only through the IUI.

```
node.name                                USER AUTHORIZATION                                hh:mm
CMD ==>

      AUTH  COMMAND                        AUTH  COMMAND
-----  -
1) YES - CHANGE PROCESS                  14) YES - SELECT STATISTICS
2) YES - DELETE PROCESS                  15) YES - SELECT TASK
3) YES - DELETE TYPE                     16) YES - SELECT TYPE
4) YES - DELETE USER                     17) YES - SELECT USER
5) YES - FLUSH PROCESS                   18) YES - SUBMIT PROCESS
6) YES - FLUSH TASK                       19) YES - SUSPEND PROCESS
7) YES - INSERT/UPDATE TYPE              20) YES - SQL SELECT
8) YES - INSERT/UPDATE USER             21) YES - SQL INSERT
9) YES - MODIFY (TRACE)                  22) YES - SQL DROP
10) YES - STOP CONNECT:Direct            23) YES - SQL CREATE
11) YES - UPDATE NETWORK MAP             24) YES - SQL GRANT
12) YES - SELECT NETWORK MAP             25) YES - STAT COMMAND
13) YES - SELECT PROCESS                  26) YES - EVENT COMMAND
```

Note: Enter the corresponding number on the command line to branch to an authorized function.

The User Authorization screen displays your authorization level for specific CONNECT:Direct commands. The following table defines the meaning of each authorization level.

Authorization Level	Definition
Yes (Y)	indicates you are authorized to perform the command
No (N)	indicates you are not authorized to perform the command
SUB	indicates you are authorized only if you submitted the Process

Obtaining Online Help

CONNECT:Direct provides online help to describe the parameters you can enter through the IUI. You can access the on-line help in two ways:

- ▶ Type **HELP** on the command line and press **ENTER**
- ▶ Press the **PF1** key

Navigating Through The Help Screens

Use the following navigation keys to guide you through the CONNECT:Direct help screens.

Navigation Key	Description
ENTER	Use to scroll through the help screens
PF1	Indicates to continue to the tutorial
PF3	Returns you to the screen where you requested help

Moving Through the Interactive User Interface

The IUI provides various ways for you to move from screen to screen through branching commands and function keys.

Branching from the Command Line

You do not need to return to the Primary Options Menu or the Administrative Options Menu to branch from one CONNECT:Direct data entry screen to another CONNECT:Direct data entry screen. You can use the command line to branch to other screens.

Type the branching command on the command line (**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 branch 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.

Branching to an Administrative Screen

If you are branching to an administrative screen from a primary screen, you must type =**ADMIN.xx** where xx is the screen ID.

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.

Branching to ISPF

The SPF option on the Primary Options Menu branches to the ISPF session, one level beneath the CONNECT:Direct panels. If you want to bypass the ISPF menu, follow the SPF command with a period and the option.

For example, to use the BROWSE function within ISPF, 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

(continued)

Key	Result
PF10	Scrolls right
PF11	Scrolls left

Branching from the IUI Summary Displays

The CONNECT:Direct IUI provides you with the option of requesting summary displays, such as the Operator Table of the SELECT PROCESS command (SP) or the Statistics Summary of the SELECT STATISTICS command (SS). These IUI summary displays allow you to branch to detail screens which further describe the summary record. Type an **S** to the left of the name of the summary record.

Browsing the IUI Displays

The TABLE or DISPLAY parameter specifies 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. You can browse the output through the use of such commands as UP, RIGHT, or FIND. Press the **PF1** key to see extensive help on how to browse the file.

Operator Tables

A feature that is available only under the IUI is the Operator Table. Use the Operator Table to monitor and control Processes. Access the Operator Table through the SELECT PROCESS command. The Operator Table provides a summary of PNODE Processes in the TCQ. See page 8-20 for a description and samples of the Operator Table.

node.name		OPERATOR TABLE			ROW 1 TO 5 OF 5	
==> Q					SCROLL ==>	PAGE
OPTION	PNAME USERID	PNUMBER	SUBMITTER.NODE	OTHER.NODE	QUEUE	
-	DELPROC USER01	100	CD.DALLAS	CD.LOSANGELES	TI	HR
R	HLDPROC USER02	101	CD.DALLAS	CD.NEWYORK	TI	WC
-	RELPROC USER01	102	CD.DALLAS	CD.LOSANGELES	HO	HI
-	SELPROC USER01	105	CD.DALLAS	CD.BOSTON	HO	HR
-	SUSPROC USER03	104	CD.DALLAS	CD.NEWYORK	EX	EX

Monitor the progress of the Processes displayed on the screen by pressing the **ENTER** key to refresh the display.

You can **Hold**, **Delete**, **Release**, suspend (**P**), or Flush Processes. (The **Show** detail option is also available as described on page 4-9.) Type the appropriate option (**H**, **D**, **R**, **P**, **F**, or **S**) in the **OPTION** column next to the name of the Process as shown in the previous figure.

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.

Viewing Messages

You may encounter two types of messages: ISPF messages and CONNECT:Direct messages.

- ▶ ISPF messages display at the top of the screen when the value you enter for a field does not pass the editor. In this example, a user typed an alphabetic character in a numeric field, Process Number, resulting in the message *PROCESS NUMBER* in the top right corner. Pressing **PF1** displays the long message *"ABC" IS INVALID...*

```
node.name          SELECT PROCESS          PROCESS NUMBER
CMD ==>
"ABC" IS INVALID; MUST BE IN THE RANGE 0 - 199,999.          hh:mm
                                                           CMD: OPR

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

PROCESS NUMBERS:
==> ABC _____ ==> _____ ==> _____
PROCESS NAMES:
==> _____ ==> _____ ==> _____
STATUS: (HO,HR,HI,HE,HC,HP,HS,RH,RA,WC,H,R,W)
==> _ ==> _ ==> _
DESTINATION NODES:
==> _____ ==> _____
==> _____ ==> _____
USER ID:          NODE ID:
==> _____
==> _____
==> _____

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

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

- ▶ CONNECT:Direct messages contain a short and long text which explain any error you get when you submit your command. For example, a user enters an invalid Process name on a SELECT PROCESS command.

The following example illustrates a short text message.

```
No process(es) found matching the search criteria.
```

Long text messages provides a detailed explanation of the message and often includes 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 Statistics Summary screen 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 has failed (indicated by an *****) and CONNECT:Direct displays the extended message.

The following example illustrates a long text message.

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

MSGID ==> SOPA006I          MODULE ==> DMDSPPRC

No process(es) found matching the search criteria.

There were no processes matching the criteria on the queues
searched.

SYSTEM ACTION: Return to invoker with RC=8.
RESPONSE: Check search criteria and retry
```

Viewing or Printing CONNECT:Direct Message Text

You can use the Message Maintenance screen to view or print Message Text. To access the screen from the Primary Options Menu select option **MSG**. The MSG command is available only through the IUI.

```
CONNECT:Direct (TM) MESSAGE MAINTENANCE          DATE => yyyy/mm/dd
ENTER OPTION ==> _          TIME => hh:mm
=====
                        OPTIONS
1 = DISPLAY MESSAGE TEXT BY MESSAGE ID
2 = DISPLAY LIST OF ENTRIES (OPTIONALLY BY MODULE)
   ==> _____ (MODULE NAME)
3 = PRINT SUMMARY REPORT
4 = PRINT FULL REPORT
```

Make your selection by typing in one of the four options on the command line and pressing **ENTER**.

Message Display Screen

This example illustrates Option 1 from the CONNECT:Direct Message Maintenance screen. You might use this option to look up an error message shown on your Process statistics or if you have a batch job that failed.

Type the message ID and press **ENTER**. An example follows.

```
CONNECT:Direct (TM) MESSAGE DISPLAY
MESSAGE ID==> SCBI190I    ISSUING MODULE==> DMCBSUBM    DATE => yyyy/mm/dd
                                     TIME => hh:mm
=====
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 ==>
LINE 6 ==> 2) The process library has not been properly concatenated.
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
```

Type **DIR** at the **COMMAND** prompt at the bottom of the screen to display the directory of the module's messages, as shown in the figure in the following section.

Message Summary Screen

The following figure illustrates Option 2 from the Message Maintenance screen. It provides a list of all the messages generated by the module you select, showing the module name and short text. If you do not specify a module name, all messages are displayed.

```
-----CONNECT:Direct MESSAGE SUMMARY----- ROW 1 to 21 OF 7748
CMD ==> SCROLL ==> PAGE
MSG ID  MODULE  MESSAGE
AAPR100I AP_CHECK Call to AP_PARSE_OUT failed.
AAPR101I DTF_AP_C NDM-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.
AAPR519I AP_PARSE Unknown keyword.
AAPR521I AP_PARSE No value specified for PID.
AAPR522I AP_PARSE No value specified for SEQ.
```

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

Message Print Screen

Select Option 3 – **Print Summary Report** or Option 4 – **Print Full Report** from the Message Maintenance screen to print the messages file in summary form or in detail, including the long form of the message text.

Provide 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, provide the name of the data set. The data set name must be enclosed in single quotes and have the following DCB attributes.

```
DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=3120
```

Issuing Commands Through the Command Line Interface

This chapter provides an overview of the CMS Command Line Interface. You issue commands through the CMS Command Line Interface (CLI) in the same way that you issue commands in the batch environment.

You request CONNECT:Direct services in the command line environment through the CMS Command Line Interface program, DMCHLAPI. The CMS Command Line Interface communicates with the DTF running under GCS. Submit a job stream that contains the CONNECT:Direct commands as SYSIN input. DMCHLAPI reads the input data stream and processes the requested functions. DMCHLAPI supports all CONNECT:Direct commands.

CMS Command Line Interface File Definition

Before using the CLI, these files must be defined:

- ▶ 'FILEDEF DMSIGNON DISK *fn ft fm*' where the *fn ft fm* is a file containing a valid CONNECT:Direct signon command. The actual *fm* must be specified.

Note: There should be no continuation marks in this file. The SIGNON command should be placed on one line and must begin *after* column 1. The GCSAPI must be on the same line as the SIGNON command.

- ▶ 'FILEDEF DMPRINT T' (optional)

A GLOBAL LOADLIB command must be issued for the library containing CONNECT:Direct modules.

Using the CMS Command Line Interface

Enter NDM, followed by any valid CONNECT:Direct command to use the CMS command line interface.

An example of the CMS command line interface follows:

```
NDM SEL STAT WHERE (STARTT=(TODAY,9AM))
```

Using the Select Message Command Through the Command Line Interface

The ability to view CONNECT:Direct messages by issuing the SElect MSG command is available through the CMS Command Line Interface.

An example of using the SEL MSG command through the CMS command line interface follows:

```
EXECOS NDM SEL MSG WHERE (ID=messageid) LONG | SHORT
```

The Select Message command allows you to specify the message to display and whether to display the LONG or SHORT form of the message. If LONG or SHORT are not specified, the SHORT form is the default.

The Select Message command can be used within a REXX EXEC. In the following example, the variables UIRTNCD, UIMSGID, UIPROC#, and UIUNODE are used. These values are taken from a CONNECT:Direct control block called the UICB (User Interface Control Block). UIRTNCD and UIMSGID are the return code and message ID returned from the CONNECT:Direct command just issued.

UIPROC# is the CONNECT:Direct Process number, resulting from a SUBMIT and UIUNODE is the CONNECT:Direct node where the Process was submitted. These values are available within an EXEC because they have been matched up with REXX variables of the same name by the CONNECT:Direct command line interface.

The following example implements the Select Message command. A Process is submitted and the return code (UIRTNCD) is checked. If it is not equal to zero, then the SEL MSG command is issued to display the error message. Otherwise, the Process number (UIPROC#) and node (UIUNODE) are displayed.

```
SUB EXEC
/* GET TEXT OF MESSAGE */
ARG PROC
"EXECOS NDM SUBMIT DSN='proc" NDMPROC A'"
IF UIRTNCD - = 0 THEN
'EXECOS NDM SEL MSG WHERE(ID='uimsgid') LONG'
ELSE
SAY "PROCESS # " uiproc# " SUBMITTED TO " uiunode
```


Managing Sessions

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

- ▶ Signing on to CONNECT:Direct
- ▶ Setting up signon defaults for the IUI
- ▶ Signing on to multiple sessions
- ▶ Swapping nodes
- ▶ Signing off 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 to illustrate the batch interface. If you type **NDM** before each batch interface example, the command is sent through the CMS Command Line Interface (CLI).

Using the SIGNON Command

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

Access to the CONNECT:Direct system can be controlled through the CONNECT:Direct Authorization Facility and through security exits. Refer to the *Controlling Security* chapter in the *CONNECT:Direct for VM/ESA Administration Guide* for information on security control.

SIGNON Command Format

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	GCSAPI = <u>userid</u> for GCSAPI
		CASE = Yes <u>No</u>
		FOLD = Yes <u>No</u>
		NETMAP = <u>network map data set name</u>
		NODE = <u>nodename</u>
		PACCT = ' <u>pnode accounting data</u> '
		RECONNECT
		SACCT = ' <u>snode accounting data</u> '
		TMPDSN = <u>temporary data set name</u>
		USERID = (<u>id</u> , <u>pswd</u> , <u>newpswd</u>)

Required Parameter

The following is the required parameter for the SIGNON command.

GCSAPI=userid for GCSAPI

specifies the VM userid where the GCS API is running.

Optional Parameters

The following parameters, listed alphabetically, are optional.

CASE = Yes | No

specifies case sensitivity for parameters associated with accounting data, userid, password, and data set name in the commands which follow the signon command. The default is No.

You can override the parameter on a command-by-command basis by adjusting the case sensitivity parameter on the individual command.

FOLD=Yes | No

determines printing uppercase control. It specifies whether printed output, an option of the various SELECT commands, is all uppercase letters. The default value is

No, specifying output printed in uppercase and lowercase letters.

NETMAP=network map data set name

specifies the CONNECT:Direct Network Map file name, which is a 1–44 character alphanumeric string with the first character alphabetic. NETMAP is required ONLY if it is not specified in a DLBL statement within the startup EXEC, usually named PROFILE GCS, of the DTF to which you are signing on. If the NETMAP is specified, it must match the filename specified on NETDSN within the user supplied INITPARM file of the targeted DTF. Otherwise, a CONNECT:Direct License Violation will be detected.

Note: If the DMNETMP ddname is provided in the job stream used with DMBATCH, this parameter is not required.

NODE=nodename

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.

PACCT='pnode accounting data'

specifies the accounting data for the primary node (PNODE). The maximum length of the accounting data is 256 characters. If special characters, such as a space, are part of the accounting data, you must enclose the string in single quotation marks. CONNECT:Direct uses this data as a default for each Process unless you override it on the SUBMIT command, PROCESS statement, or SUBMIT statement.

When using the IUI, if you do not specify a destination, CONNECT:Direct routes the output the SYSOUT class specified in the initialization parameters (external to CONNECT:Direct) under the TSO userid. 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.

RECONNECT

specifies that you can re-establish your VTAM session without having to sign off and sign on again. You might lose a VTAM session due to a link or line failure.

The USERID parameter is the only other parameter that you should specify with the RECONNECT option.

SACCT='snode accounting data'

specifies the accounting data for the secondary node (SNODE). 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.

USERID=(id, pswd, newpswd)

specifies your signon security information.

id specifies the security ID that CONNECT:Direct will pass to a security exit. The range is 1–8 characters.

pswd specifies the current security password. The security exit uses this parameter to validate the current security password. It contains 1–8 alphanumeric characters.

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 1–8 alphanumeric characters.

When you attempt a sign on to a remote node, the userid and password information must correspond to security procedures at that remote node.

Using SIGNON Through the Batch Interface

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

1. Place the SIGNON command in the DMBATCH job stream as described in the *Issuing Commands Through the Batch Interface* chapter beginning on page 3-1.
2. Submit the Process while CONNECT:Direct is running.
3. Verify the results.

Batch Interface SIGNON Example

The following example shows a SIGNON command in which the user specifies accounting data for the PNODE and SNODE.

```
SIGNON  NETMAP = DALLAS.NETMAP -
        USERID = (MYUID1,MYPSWD) -
        GCSAPI = Y123490 -
        PACCT  = 'JOB FOR SYSMAINT, DEPT.27, MARKETING' -
        SACCT  = 'JOB FOR INVENTORY, DEPT.55, ACCOUNTING'
```

Type **NDM** before the SIGNON command in the previous example to submit the job using the CMS Command Line Interface (CLI). For more information on the CLI, refer to the *Issuing Commands Through the Command Line Interface* chapter beginning on page 5-1.

See page 6-14 for more Batch Interface examples of the SIGNON command.

Using SIGNON Through the IUI

To issue the SIGNON command through the CONNECT:Direct IUI, perform the following steps:

1. Select the CONNECT:Direct option from your ISPF Primary Option Menu. You will see the Copyright screen and the Signon Screen.
2. Type your userid (id, password, and new password), node name, and case sensitivity option and press **ENTER**. The parameters of the SIGNON command are described beginning on page 6-2.
3. Verify your SIGNON.

The message *Error requesting session, CONNECT:Direct may be inactive* indicates that the CONNECT:Direct DTF has not been started. If your signon is successful, the Primary Options Menu appears. Select a command screen from the Primary Options Menu.

Viewing Your Current Signon Parameters

To view your current signon parameter defaults, complete these steps:

Select the **WHO** option from the Primary Options Menu to branch to the User Information display. Press **PF1** for online descriptions of the various fields.

```
CONNECT:Direct USER INFORMATION                hh:mm
CMD ==>
USER ID                ==>  USER01
USER NODE              ==>  CD.SOMERSET
GCS API ID            ==>  CDAPP01
VTAM APPLID          ==>  CIAPP01
TEMPORARY DSN DDNAME ==>  $000003

DATA SETS:
TEMPORARY DSN        ==>  NDMAPI LISTING
NETWORK MAP DSN     ==>  CD.NETMAP.NER01

IUI VERSION          ==>  03
IUI RELEASE          ==>  002
IUI MOD-LEVEL       ==>  000
IUI PUF-LEVEL       ==>  000
IUI PUT-LEVEL       ==>  0000
```

Setting up Signon Defaults

To change the SIGNON parameters, perform these steps:

1. Select **SD** from the Primary Options Menu to branch to the SIGNON DEFAULTS display.

```
node.name                SIGNON DEFAULTS                hh:mm
CMD==>
CURRENT DEFAULTS:
USER ID                ==>  USER01
PASSWORD              ==>
USER NODE NAME        ==>  CSDAPP03
VM ID                 ==>
EXTENDED SUBMIT FEATURE ==>
UPPER CASE FOR PRINT  ==>
ENODE ACCOUNTING DATA ==>
SNODE ACCOUNTING DATA ==>
DO YOU WANT ALL COMMANDS FOR THIS SESSION TO BE CASE SENSITIVE? ==> NO
INITIAL COMMAND       ==>
```

2. Change the appropriate fields to modify your signon defaults. See page 6-2 for SIGNON command parameters valid values for the options you can modify.

The INITIAL COMMAND field allows 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).

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.

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 as follows.

Label	Command	Parameters
(optional)	SIGNON	NODE = nodename
		PACCT = 'pnode accounting data'
		RECONNECT
		SACCT = 'snode accounting data'
		USERID = (id, pswd, newpswd)

See page 6-2 for SIGNON command parameters valid values for 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	Parameter
(optional)	SIGNON	NODE = nodename
		RECONNECT
		USERID = (id, pswd, newpswd)

See page 6-2 for SIGNON command parameters valid values for these parameters.

SIGNOFF Sequence

Observe the following rules to control your SIGNOFF command sequence:

- ▶ Always remember which session is the master session (the first node you signed on to in this session). 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. This occurs regardless of the order in which the sessions were established.

See the *Sequencing the SIGNON, SWAP NODE, and SIGNOFF Commands* section starting on page 6-14 for examples of the importance of the sequence of your SIGNOFF commands.

Switching Nodes

When you issue multiple signons you may want one node to perform a function and another node to perform a different function. The SWAP NODE command allows you to swap from node to node. See page 6-10 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, perform the following steps:

1. Place the multiple session SIGNON command in the DMBATCH job stream as described in the *Issuing Commands Through the Batch Interface* chapter beginning on page 3-1.
2. Submit the Process while CONNECT:Direct is running.
3. Verify the results.

See the examples beginning on page 6-14 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.

Note: Type **NDM** before the SIGNON command in the examples to submit the job using the Command Line Interface (CLI). Refer to the *Issuing Commands Through the Command Line Interface* chapter beginning on page 5-1 for more information.

Using the IUI Multiple Session SIGNON Command

The Multiple Session Signon screen allows you to sign on to more than one node during a single ISPF session.

The communications name (APPLID) value in the Network Map definition determines the maximum number of sessions that you can sign on to during an ISPF session. Each signon uses an additional VTAM APPLID.

Instructions for Using Multiple Session SIGNON Through the IUI

To issue multiple session SIGNON through the CONNECT:Direct IUI, perform the following steps:

1. Access the Multiple Session Signon screen by selecting **MS** from the Primary Options Menu. The following example illustrates the Multiple Session SIGNON display.

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

                                MULTIPLE SESSION SIGNON

USER ID ==> userid1
PASSWORD ==>
NEW PASS ==>
NODE NAME ==> node.name1
RECONNECT ==> (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
```

2. Type your userid (id and password), node name, and reconnect option and press **ENTER**.

Refer to the descriptions of the SIGNON command parameters beginning on page 6-2 for valid values for the fields displayed on the screen or press the **PF1** key for online help.

Swapping Nodes

The SWAP NODE command allows 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 to which node will issue a command.

SWAP NODE Command Format

The only parameter of the SWAP NODE command is the **nodename** parameter which specifies which node you will swap to in a multiple session environment.

Note: The nodename is the name assigned to the LOCAL.NODE field defined in the Network Map definitions.

Using SWAP NODE Through the Batch Interface

To issue this command from the batch interface, perform these steps:

1. Place the SWAP NODE command in the DMBATCH job stream as described in the *Issuing Commands Through the Batch Interface* chapter beginning on page 3-1.
2. Submit the Process while CONNECT:Direct is running.
3. Verify the results.

For example, this command swaps you from your current active remote node (userid of JONES2) to the local node under the userid 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
/*
```

Note: Type **NDM** before the SWAP NODE command in the example to submit the job using the Command Line Interface (CLI). Refer to the *Issuing Commands Through the Command Line Interface* chapter beginning on page 5-1 for more information.

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. To issue SWAP NODE through the CONNECT:Direct UI, perform the following steps:

1. Access the SWAP/DISPLAY USER SESSIONS screen by selecting option **SW** from the Primary Options Menu.

The following example illustrates the SWAP/DISPLAY USER SESSIONS display. The screen displays all user sessions for the current ISPF session.

```

-----SWAP/DISPLAY USER SESSIONS----- ROW 1 TO 2 OF 2
==>                                     SCROLL ==> PAGE
  NODE                                APPLID    CURRENT    STATUS
-----
  node.name1                          applid0    *ACTIVE*   *MASTER* **LOCAL*
                                userid2
  node.name3                          applid1
                                userid1
***** BOTTOM OF DATA *****

```

2. Type **S** to swap sessions or **D** to delete or sign off a session in the input field to the left of the node name. Refer to the next two sections for more information about swapping and deleting sessions.

Swapping Sessions

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.

Deleting Sessions

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. Refer to page 6-8 to see the the significance of your SIGNOFF command sequence.

Signing Off CONNECT:Direct for VM/ESA

The SIGNOFF command terminates a connection to the DTF.

SIGNOFF Command Format

The SIGNOFF command has the following format. There are no associated parameters.

Label	Command	Parameters
(optional)	SIGNOFF	

See the information on page 6-8 concerning the significance of your SIGNOFF command sequence if you are using the Multiple Session Signon feature.

Using SIGNOFF Through the Batch Interface

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

1. Place the SIGNOFF command in the DMBATCH job stream as described in the *Issuing Commands Through the Batch Interface* chapter beginning on page 3-1.
2. Submit the Process while CONNECT:Direct is running.
3. Verify the results.

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

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

You can type **NDM** before the SIGNOFF command in the example to submit the job using the Command Line Interface (CLI). Refer to the *Issuing Commands Through the Command Line Interface* chapter beginning on page 5-1 for more information.

See the examples in the *Sequencing the SIGNON, SWAP NODE, and SIGNOFF Commands* section on page 6-14 for more examples of the SIGNOFF command used with the Multiple Session Signon feature and the SWAP command.

Using SIGNOFF Through the IUI

The following table lists 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	Command Line

When you SIGNOFF, **CONNECT:Direct** returns to the ISPF menu.

Sequencing the SIGNON, SWAP NODE, and SIGNOFF Commands

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

Example 1: Terminating Two Sessions Through One SIGNOFF

The following example shows a SIGNON to two different nodes and the use of the SWAP NODE command.

```
/* 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
```

Note: The SIGNOFF command shown in this example causes **CONNECT:Direct** to terminate both sessions, because the active session at signoff was the master session.

Example 2: Terminating a Master Session with SIGNOFF Command

In the following example, the first SIGNOFF command will terminate the CD.NEWYORK node because it was the last SIGNON command issued.

```
/* 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
```

Note: 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.

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 were signed on.

```
/* 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
```

If you try to swap to a node that is already active, CONNECT:Direct ignores the SWAP NODE command. The USERID (or any other userid 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.

Building, Modifying, and Submitting Processes

This chapter provides the information you need to submit Processes to the Transmission Control Queue (TCQ) using the SUBMIT command.

Note: The SUBMIT command does not submit a job to the VM job entry subsystem. It submits your Process to the TCQ.

Review the *CONNECT:Direct for VM/ESA Administration Guide* for more information on how Processes move through the TCQ.

You can issue the SUBMIT command when you are:

- ▶ Submitting an existing Process using the Batch Interface
- ▶ Submitting a Process through the IUI

Refer to the *CONNECT:Direct Process Guide* for instructions on building a Process and the *Controlling Processes in the TCQ* chapter beginning on page 8-1 of this book for how to control Processes in the TCQ.

Note: To monitor a Process while it is in the TCQ, use the SELECT PROCESS command. To determine the outcome of a finished Process, use the SELECT STATISTICS command.

Each section describes the command format and parameters for the batch and Interactive User Interface. Examples are given in the batch sections to illustrate the batch interface. If you type **NDM** before each batch interface example, the command is sent through the CMS Command Line Interface (CLI).

Submitting a Process Using the SUBMIT Command

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

SUBMIT Command Format

The SUBMIT 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)	SUBmit	PROC = <u>filename</u>
		DSN = 'fn ft [fm] [(member)]
		CASE = Yes No
		CLASS = n
		HOLD = Yes <u>No</u> Call
		MAXDELAY = [<u>UNLIMITED</u> <u>QUEUED</u> 0 hh:mm:ss]
		NEWNAME = newname
		NOTIFY = %USER userid
		PACCT = 'pnode accounting data'
		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=tcvalue
		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

Required Parameters

The only parameter required for the SUBMIT command is the one that identifies the Process. Use the PROC parameter if the Process resides in

the CONNECT:Direct Public Process Library. Use the DSN parameter if it does not. Either PROC or DSN must be specified; *do not specify both*.

PROC=filename

specifies the Process name. The Process name is a 1–8 character alphanumeric string, with the first character alphabetic. Either PROC or DSN must be specified; do not specify both. The Process has a filetype of NDMPROC. The Process that will be submitted is 'filename NDMPROC'.

DSN=' fn ft [fm] [(member)]'

specifies the name of the file that contains the Process. Either PROC or DSN must be specified; do not specify both. The DSN parameter can be used to specify the filename (fn), filetype (ft), optional filemode (fm), and optional member name if the Process resides in a MACLIB.

Note: If the Process has been edited, the DTF machine must reaccess the disk where the Process resides.

Optional Parameters

The following are the optional parameters for the SUBMIT command.

CASE = Yes | No

specifies if parameters associated with accounting data, userid, password, and data set name in the command *and in the Process* are to be case sensitive. See page 2-9 for more information on how this parameter functions.

CLASS = n

determines the node-to-node session on which a Process can execute. Consult the CONNECT:Direct administrator for instructions concerning which class to specify. The range is 1–255.

HOLD = Yes | No | Call

specifies if the Process will be placed in the hold queue.

Yes specifies that the Process will remain in the hold queue until one of the following events occurs:

- A CHANGE PROCESS command releases the Process
- A DELETE PROCESS command deletes the Process

When you specify both HOLD=YES and a STARTT value, HOLD takes precedence. CONNECT:Direct places a

Process submitted with HOLD=YES in the hold queue even if you specified a start time.

No specifies that CONNECT:Direct will not place the Process in the hold queue, but will place it in the WAIT for execution queue (EX). HOLD=NO is the default.

Call specifies that CONNECT:Direct will place the Process in the hold queue until a VTAM 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 with HOLD=NO establishes a session and causes execution of any Processes residing on the SNODE destined for this node that were submitted with HOLD=CALL.

Note: CONNECT:Direct ignores the HOLD parameter if RETAIN=Y.

MAXDELAY = [UNLIMITED | QUEUED | 0 | hh:mm:ss]

indicates that the **submit** command waits until the submitted Process completes execution or the specified time interval expires.

unlimited specifies the **submit** command to wait for the Process to complete execution. UNLIMITED is the default if MAXDELAY is coded without any parameters.

queued specifies the **submit** command waits until the process completes or 30 minutes, whichever occurs first.

0 specifies the **submit** command will attempt to start a session for the submitted Process to execute on immediately. The submit command waits until the Process completes or until all timer retries have been exhausted.

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.

NEWNAME = newname

specifies the new name of the Process. The default is the label on the Process statement.

NOTIFY = %USER | userid

specifies the userid to receive Process completion messages.

%USER specifies that the user who submitted the Process will receive the completion messages, if the **CONNECT:Direct** userid that the user is currently logged on with is the same as the VM userid.

Note: If the **CONNECT:Direct** userid is different from the VM userid, the user will not be notified.

userid specifies the VM userid that will receive Process completion messages.

PACCT = 'pnode accounting data'

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 data overrides any accounting data specified on the **SIGNON** command, as well as any accounting data specified in the Process statement of the submitted Process.

PNODE = primary node name

specifies the primary node of the Process.

primary node name is a 1–16 alphanumeric character name defined in the network map. The name can be expressed in alphanumerics or nationals (@#\$) with embedded periods.

The node to which you submit the Process is always the PNODE. This parameter defaults to the name of the node submitting the Process and need not be specified. PNODE is for documentation purposes only. For information on node name formats and values, see the SNODE parameter on page 7-7.

PNODEID = (id, pswd, newpswd)

specifies security userids and security passwords at the PNODE.

id specifies the security ID that **CONNECT:Direct** passes to a security exit for validations on the PNODE side of the Process. It can contain 1–64 alphanumeric characters.

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.

newpswd specifies the new security password. The security exit uses this parameter to change the current security password to the new security password. It can contain 1–64 alphanumeric characters.

PRTY = n

specifies the Process priority in the TCQ. 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 the priority defined during installation. The range is from 0–15.

REQUEUE = Yes | No

specifies whether CONNECT:Direct should requeue a COPY step if an abend occurs during processing.

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).

No specifies the failing copy step is not requeued if it fails with an abend. 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.

RETAIN = Yes | No | Initial

specifies whether or not CONNECT:Direct keeps a copy of the Process in the HOLD queue after the Process has executed.

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.

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.

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.

SACCT = 'snode accounting data'

specifies the accounting data for the SNODE. 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 data overrides any accounting data specified on the SIGNON command, as well as any accounting data specified in the Process statement.

SNODE = secondary node |

SNODE=TCPNAME = tcpvalue

specifies the secondary node to be used in this Process.

secondary node name is a 1–16 alphanumeric character name that is defined in the Network Map. The name can be expressed in alphanumerics or nationals (@#)\$ with embedded periods.

Use the **TCPNAME=tcpvalue** form of the SNODE parameter to specify TCP/IP connections that are not defined in the CONNECT:Direct Network Map.

tcpvalue is defined as the TCP/IP network address, the network name, or an alias for the network name. The range is from 1–16 characters with embedded periods. If the network name is longer than 16 characters, you must specify an alias for the network name.

If the TCPNAME keyword is used, the default TCP/IP port number is assumed.

SNODEID = (id, pswd, newpswd)

specifies security userids and security passwords at the SNODE.

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.

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 an SNODE userid override is specified. The actual generation depends on the information shared in the PNODE Authorization File and the option used to generate the stage 2 security exit. The *CONNECT:Direct for VM/ESA Administration Guide* contains details for the Authorization File and Stage 2 security exit.

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.

STARTT = ([date | day] [,hh:mm:ssXM])

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.

Do not code the STARTT parameter with the RETAIN=I parameter.

date specifies the date from which the statistics records are selected. CONNECT:Direct is Year 2000 compliant. For detailed information on Year 2000 compliance, refer to the *CONNECT:Direct for VM/ESA Release Notes*.

Note: STARTT should not be coded with RETAIN=INITIAL.

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 date values.

Note: You can omit the separators only for transfers between mainframe nodes. Sterling Commerce suggests the use of separators to guarantee transfers between all platforms.

After you designate the date order in your initialization parameters, you can use the following date formats:

If you specify DATEFORM=MDY, CONNECT:Direct interprets the date format as:

- ▶ mm/dd/yy or mm/dd/yyyy
- ▶ mm.dd.yy or mm.dd.yyyy

If you specify DATEFORM=DMY, CONNECT:Direct interprets the date format as:

- ▶ dd/mm/yy or dd/mm/yyyy
- ▶ dd.mm.yy or dd.mm.yyyy

If you specify DATEFORM=YMD, CONNECT:Direct interprets the date format as:

- ▶ yy/mm/dd or yyyy/mm/dd
- ▶ yy.mm.dd or yyyy.mm.dd

If you specify DATEFORM=YDM, CONNECT:Direct interprets the date format as:

- ▶ yy/dd/mm or yyyy/dd/mm
- ▶ yy.dd.mm or yyyy.dd.mm

CONNECT:Direct processes Julian dates the same as previous releases. The only change in Julian date processing is the provision for four-digit years. The following formats are valid:

- ▶ yyddd or yyyyddd
- ▶ yy/ddd or yyyy/ddd
- ▶ yy.ddd or yyyy.ddd

If only date is specified, the time defaults to 00:00.

Note: If RETAIN=YES, a date cannot be specified in the STARTT parameter.

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 will release the Process for execution today, or TOMORROW, which will release the Process for execution the next day.

Note: If you specify the day of the week with RETAIN=YES, the Process will execute the same day every week.

If you do not specify the time, it will default to 00:00. This means that if you submit a Process on Monday, with Monday as the only STARTT parameter, the Process will not run until the following Monday.

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.

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.

Note: If you code hh:mm:ssXM with RETAIN=YES, CONNECT:Direct schedules the Process for execution the same time every day.

You can also specify NOON, which will release the Process for execution at noon, or MIDNIGHT, which will release the Process for execution at midnight.

Note: 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 specify a start time.

&symbolic 1 = variable string 1

&symbolic 2 = variable string 2...

&symbolic N = variable string N

specifies the default value substituted for the symbolic parameter in the Process. A symbolic overrides any default values you specify in the Process statement.

Specify a null value by immediately following the equal sign with a comma. 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 was resolved during the first process submission. Do not use identical symbolic names. The maximum variable string length is 256 characters.

Using SUBMIT Through the Batch Interface

To issue this command from the batch interface, perform these steps:

1. Place the SUBMIT command in the DMBATCH job stream as described in the *Issuing Commands Through the Batch Interface* chapter beginning on page 3-1.
2. Submit the Process while CONNECT:Direct is running and verify the results.

Batch Interface SUBMIT Example

The following example shows a SUBMIT command for the Process named PAYROLL. CONNECT:Direct puts this Process on the HOLD queue and retain a copy after it is released for execution.

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

This example shows a SUBMIT command for the Process named COPYSEQ. CONNECT:Direct resolves the symbolic &DSN in the Process COPYSEQ upon submission.

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

Using SUBMIT Through the IUI

Processes may be submitted three ways through the IUI.

- ▶ **SB** Submit Process
- ▶ **DF** Define Process
- ▶ **CF** Copy File

A definition for each IUI SUBMIT option follows.

Submit Process (SB)

use to submit Processes from the CONNECT:Direct Public Process Library or a library you specify on the screen. You may also override certain parameters in the existing Process. Refer to the *Submitting a Predefined Process Using the SB IUI Option* section beginning on page 7-13 for more information on this IUI SUBMIT option.

Define Process (DF)

allows you to edit, create, and submit Processes residing in the CONNECT:Direct Public Process Library. The ISPF editor is used to edit or create the Process. Refer to the *Defining or Modifying a Process Using the ISPF Editor (DF IUI Option)* section beginning on page 7-14 for more information on this IUI SUBMIT option.

Copy File (CF)

allows you to dynamically build a COPY Process through fill-in-the-blank panels and submit it. Refer to the *Generating a COPY Process and Submitting it Using the CF IUI Option* section beginning on page 7-17 for more information on the CF SUBMIT option.

Submitting a Predefined Process Using the SB IUI Option

To submit a predefined process, perform the following steps:

1. Select option **SB** from the Primary Options Menu.

```
node.name          SUBMIT PROCESS          hh:mm
CMD==>            yyyy/mm/dd
                  yyyy.ddd
PROCESS NAME      ==>      (FILENAME OF PROCESS)
OR
DATA SET NAME    ==>
SECONDARY NODE   ==> ----- TRANSPORT: NETMAP
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           ==> -----
NOTIFY          ==> %USER
SECURITY OVERRIDE ==> N
SCHEDULED       ==> - DATE - - TIME -
START           ==> ----- ==> -----
SYMBOLIC PARAMETERS ==> -----

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

2. Enter the parameters of the SUBMIT command as described on page 7-2. Press the **PF1** key for online help with field content. The case sensitivity you indicate applies not only to the SUBMIT command, but to the Process itself.

The TCPNAME keyword of the SNODE parameter described on page 7-7 is provided through the IUI by the TRANSPORT field. The TRANSPORT field determines what communication transport CONNECT:Direct will use for the file transfer.

- ▶ A value of NETMAP causes CONNECT:Direct to look in the Network Map for the sending and receiving nodes. If you enter a Transport of NETMAP, the SNODE keyword remains SNODE=secondary node.
- ▶ A value of TCP indicates that the NODE NAME field contains an alias name or an IP address. Therefore, CONNECT:Direct does not look in the Network Map for node information. If you enter TCP in the Transport field, the SNODE keyword becomes SNODE=TCPNAME=tcvalue.

If you want to override the primary or secondary node security information such as userid and password, enter a **Y** next to the Security Override prompt. The following screen displays.

```

node.name          CONNECT:DIRECT SECURITY OVERRIDE          TIME:  hh:mm
CMD==>

PRIMARY NODE:

SECURITY ID: _____
PASSWORD:
NEW PASS:

SECONDARY NODE:

SECURITY ID: _____
PASSWORD:
NEW PASS:

```

Defining or Modifying a Process Using ISPF Editor (DF IUI Option)

Use the Process Definition screen to define or update a Process using the ISPF editor and then SUBMIT the Process to the DTF. Choose option **DF** from the Primary Options Menu.

```

node.name          CONNECT:Direct          hh:mm
                   PROCESS SUBMIT/EDIT SCREEN
CMD ==>

ISPF LIBRARY:
PROJECT ==> _____
GROUP   ==> _____
TYPE    ==> _____
MEMBER  ==> _____ (Blank for member selection list)

CMS FILE:
FILEID  ==> SAMPLE CDPROC
MEMBER  ==> _____

READ PASSWORD ==> _____          UPDATE PASSWORD ==> _____

ISPF EDITOR
MESSAGE:

```

The Process Definition Screen identifies the file in which the Process resides and provides access to the ISPF editor.

How to Avoid Recovery Error Problems

With ISPF Edit Recovery (RECOVERY ON) is turned on, an ISPF message displays after modifying the member being edited.

```
ISRE306 "RECOVERY ERROR - INITIALIZATION OF BACKUP FILE FAILED, CHECK  
EDIT RECOVERY TABLE"
```

This error occurs because no Edit Recovery Table is found in the data set pointed by the ISPPROF filedef for the CONNECT:Direct application.

To create the Edit Recovery Table, enter "CMS %ISREDRTI" on the ISPF command line while you are in your current Edit session. This creates the Edit Recovery Table for the CONNECT:Direct application.

Using Models to Define a Process

To define a Process using the CONNECT:Direct models, complete the Process Definition Screen using the following steps:

1. The ISPF LIBRARY field or the CMS FILE field or both will be filled in with the file names you last entered. (The name SAMPLE CDPROC is used for this example.) Change the Process library name or the CMS file name only if you want to use another library. The Process library can be a MACLIB or a CMS set of files. The CMS file name can be a MACLIB or a sequential file.
2. Place an **S** preceding the desired member and press **ENTER** to select a member from a list, or type the member name and press **ENTER**. If the Process is a sequential file, press **ENTER**.

After you complete these steps, CONNECT:Direct invokes the ISPF editor, and the specified member or sequential file displays.

Changing an Existing Process

After you type in the member name of an existing Process, CONNECT:Direct displays the Process to you in ISPF edit mode so that you can make changes. For information on the valid content and syntax of Process statements see the *CONNECT:Direct Process Guide*. When you are finished updating, press **PF3** to save the member and return to the Process Definition screen.

Creating a New Process

If you want to create a new Process, type in the member name and press **ENTER**. Use the ISPF edit mode to enter your Process statements. For information on the valid content and syntax of Process statements see the *CONNECT:Direct Process Guide*. When you are finished, press **PF3** to save the member and return to the Process Definition screen.

Submitting the Process

To submit a Process from the *CONNECT:Direct Process Definition* Screen, perform the following steps:

1. Type **SUBMIT** on the command line.
2. Type the member name of the Process in the Process Member Name field.
3. Ensure the correct library name that contains the member to submit is displayed in either Process Library Name or Public Process Library fields. Press **ENTER**.

Using the DF Screen to Access Statement Models

Use the DF screen to access Process statement models found in the sample Process library. See the *CONNECT:Direct for VM/ESA Modal Statements* chapter of the *CONNECT:Direct Process Guide* for instructions.

Submitting a Process Using the SB IUI Option

Use the **SB** option from the Primary Options menu to access the SUBMIT PROCESS screen.

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

PROCESS NAME      ==>      (FILENAME OF PROCESS)
OR
DATA SET NAME    ==>

SECONDARY NODE    ==> _____  TRANSPORT: NETMAP
REQQUEUE         ==> -             (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            ==> _____
NOTIFY           ==> %USER
SECURITY OVERRIDE ==> N
SCHEDULED        ==>  ___ DATE ___  ___ TIME ___
START            ==> _____  ==> _____
SYMBOLIC PARAMETERS ==> _____

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

The fields on the SUBMIT PROCESS screen allow you to enter parameters of the SUBMIT command as described on page 7-2. Press **PF1** for online help with field content.

The NEW PROCESS NAME field is optional. If you do not enter a new process name, CONNECT:Direct provides the default name COPYPDS. Refer to the *CONNECT:Direct Process Guide* for the definition of the fields in the Submit Menu screens.

Generating a COPY Process and Submitting it Using the CF IUI Option

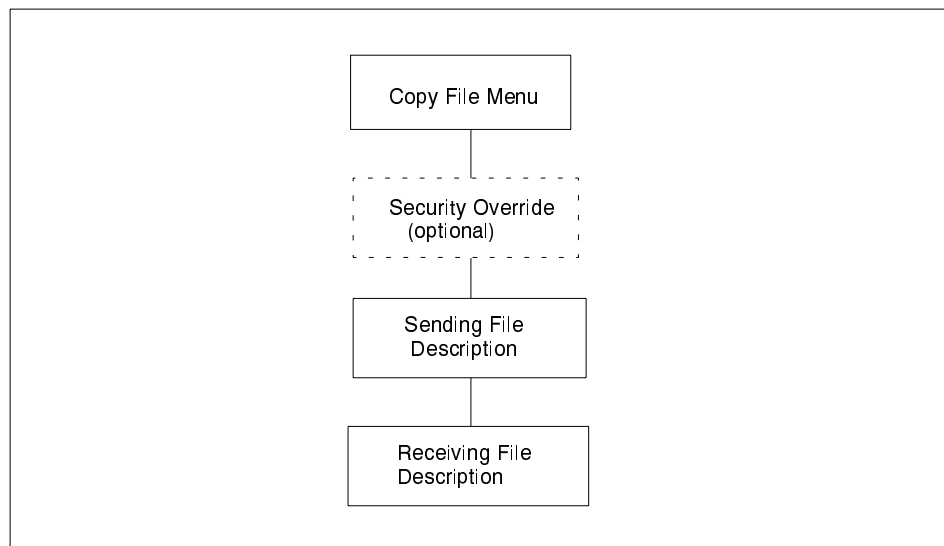
Use the Copy File Menu to generate and submit a COPY Process. Select option **CF** from the Primary Options Menu.

The Copy File menu is part of a series of panels that collect information used to build a SUBMIT PROCESS command. CONNECT:Direct submits the predefined Process COPYCF with the appropriate command parameters and variable substitutions based on your input.

There are four types of panels:

- ▶ The Menu panel collects information for the SUBMIT COMMAND parameters, PNODE/SNODE information, data compression, and checkpoint information. See the Menu panel on page 7-19.
- ▶ The Security Override panel collects information about userid, password, and accounting data. This panel displays only if you request it on the first panel. See the description of the Security Override panel beginning on page 7-20.
- ▶ A Sending File panel appropriate to the SENDING ENVIRONMENT collects information for the Process variables for the sending file.
- ▶ A Receiving File panel appropriate to the RECEIVING ENVIRONMENT is displayed to gather information for the receiving file variables.

The relationship of these four panels is shown in the following diagram:



The first panel displayed when you select the CF option follows.

```
node.name      CONNECT:Direct for VM/ESA COPY FILE MENU      TIME: hh:mm
CMD ==>

SENDING ENVIRONMENT: ___      TRANSPORT: NETMAP
NODE NAME: _____
RECEIVING ENVIRONMENT: ___
NODE NAME: _____

PROCESS NAME:  _____
CLASS:         _____ (NUMERIC)
HOLD:          N          (Y, N, OR C-CALL)
NOTIFY USERID: %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)
COMPRESS:     N          (Y, N, X=EXTENDED, X'XX', OR C'C')

OVERRIDE SECURITY: N          (Y OR N)

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

The fields displayed on the screens allow you to enter parameters of the SUBMIT command as described on page 7-2. Press the PF1 key for online help with field content. Fields displayed on the screen that are not parameters of the SUBMIT command follow.

PROCESS NAME

(optional) If you do not enter a Process name, CONNECT:Direct provides the default name *COPYCF*.

CHECKPOINT

specifies the byte interval for checkpoint support, which allows restart of interrupted transmissions at the last valid transmission point, thus avoiding the need to restart transmission from the beginning. **K** denotes thousands; **M** denotes millions. A checkpoint value of zero stops automatic checkpointing. See the *CONNECT:Direct Process Guide* for a complete description of this parameter.

COMPRESS

specifies that CONNECT:Direct is to compress the data, thus reducing the amount of data transmitted as the file is copied from one node to another. CONNECT:Direct automatically decompresses the file at its destination. The default subparameter for the COMPRESS parameter is PRIMEchar=X'40'. See the *CONNECT:Direct Process Guide* for a complete description of this parameter.

CASE SENSITIVITY

specifies case sensitivity which applies not only to the SUBMIT command, but to the COPY Process itself.

Security Information

If you want to enter security information such as userid and password, enter a **Y** at the Override Security prompt to display this screen.

```
node.name          CONNECT:Direct for VM/ESA SECURITY OVERRIDE          hh:mm
CMD==>

SENDING  ENVIRONMENT:  ____          TRANSPORT: NETMAP
NODE NAME:  _____

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

RECEIVING ENVIRONMENT:  ____
NODE NAME:  _____

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

The TCPNAME keyword of the SNODE parameter described on page 7-7 is provided through the IUI by the TRANSPORT field. The TRANSPORT field determines what communication transport CONNECT:Direct will use for the file transfer.

- ▶ A value of *NETMAP* causes CONNECT:Direct to look in the Network Map for the sending and receiving nodes. If you enter a Transport of *NETMAP*, the SNODE keyword remains SNODE=secondary node.
- ▶ A value of *TCP* indicates that the NODE NAME field contains an alias name or an IP address. Therefore, CONNECT:Direct does not look in the Network Map for node information. If you enter *TCP* in the Transport field, the SNODE keyword becomes SNODE=TCPNAME=tcvalue.

Sending and Receiving Environments

The values you place in the Sending and Receiving Node Name fields and their corresponding Environment fields determine which Sending and Receiving screens display to complete your COPY statement.

The following are valid Environment values for sending and receiving COPY files: AS400, MVS, MSP, NETWARE or NW, NT, OS2, PC, VM/ESA, VMS, VOS, VSE, TANDEM, and UNIX. For a complete description of the valid parameters of a COPY Statement and examples, see the *CONNECT:Direct Process Guide*.

Copyfile - Sending File Screens by Environment

The values you place in the Sending Node Name and Environment fields determine which Sending File screen displays. The Copyfile menu differs according to your environment. For example, this display illustrates the Sending File screen for MVS.

```
node.name          COPYFILE - SENDING FILE (MVS)          TIME: hh:mm
CMD ==>

NODE NAME: _____

SENDING DSNAME:   _____
UNIT PARAMETER:  ( _____ )
LABEL PARAMETER:  ( _____ )
VOL PARAMETER:   ( _____ )
DCB PARAMETER:   ( _____ )
TYPE KEY:        _____
MSVGP NAME:      _____

(PDS ONLY:)      REPLACE: Y (Y OR N)  ALIAS: Y (Y OR N)
SELECTION CRITERIA: ( _____ )
                  ( _____ )

EXCLUSION CRITERIA: ( _____ )
                  ( _____ )
```

In the next example, you can see how different the Sending File screen for OS/400 is from the previous MVS display..

```
node.name          COPYFILE - SENDING FILE (OS/400)      TIME: hh:mm
CMD ==>

NODE NAME:

OBJECT NAME: _____

OBJECT TYPE: F    (F - PHYSICAL, S - SPOOL, O - OBJECT, M - SINGLE MEMBER)

SYSOPTS:
_____
_____
_____
_____
```

Use the online help screens to see complete descriptions of each parameter. Refer to the *Obtaining Online Help* section beginning on page 4-7 for more information on how to access and navigate through the help screens.

Copyfile - Receiving File Screens by Environment

The values you place in the Receiving Node Name and Environment fields determine which Sending File screen displays. The Copyfile menu differs according to your environment. For example, this display illustrates the Receiving File screen for MVS.

```
node.name          COPYFILE - RECEIVING FILE (MVS)          TIME: hh:mm
CMD ==>

NODE NAME:

RECEIVING DSNAME:  _____

DISPOSITION:      ( NEW , CATLG , _____ )
UNIT PARAMETER:   ( _____ )
VOL PARAMETER:    ( _____ )
DCB PARAMETER:    ( _____ )
LABEL PARAMETER:  ( _____ )
SPACE:            ( _____ )
TYPE KEY:         _____
MSVGP NAME:       _____

SMS SYSTEMS ONLY
DATA CLASS: _____ STORAGE CLASS: _____ MANAGEMENT CLASS: _____
AVERAGE RECORDS: _____ DSN TYPE : _____
VSAM ORGANIZATION: _____ KEY LENGTH: _____ KEY OFFSET: _____
LIKE DSNAME: _____
SECURITY MODEL:   _____
GENERIC MODEL:   _____ (YES OR NO)
```

In the next example, you can see how different Receiving File screen for VM-DISK from the previous MVS example.

```
node.name          COPYFILE - RECEIVING FILE (VM-DISK)      TIME: hh:mm
CMD ==>

NODE NAME:
TYPE OF FILE:      R (D-DISK, V-VSAM, T-TAPE, R-READER, S-SFS)

----- DISK FILE INFORMATION -----
FILENAME: _____ FILETYPE: _____

(LINK INFORMATION)
USERID: _____ PASSWORD: _____ ACCESS MODE: W_ CUU: 191_

DCB PARAMETERS: ( _____ )
TYPE KEY: _____
OLDDATE:         N (Y OR N)
REPLACE FILE:    Y (Y OR N)

SELECTION CIRTERIA: ( _____ )
                  ( _____ )

EXCLUSION CITERIA: ( _____ )
                  ( _____ )
```

Use the online help screens to see complete descriptions of each parameter. Refer to the *Obtaining Online Help* section beginning on page 4-7 for more information on how to access and navigate through the help screens.

Controlling Processes in the TCQ

This chapter provides information about controlling and monitoring Processes in the TCQ. Chapter topics include:

- ▶ Controlling Processes through commands
- ▶ Modifying select parameters of a Process
- ▶ Suspending, flushing, and deleting Processes
- ▶ Examining Processes
- ▶ Tracking and controlling Processes through the Operator Table

Refer to the *CONNECT:Direct Process Guide* for instructions on building a Process and the *Building, Modifying, and Submitting Processes* chapter beginning on page 7-1 of this book for instructions on submitting the Process to the TCQ.

Controlling Processes Through Commands

CONNECT:Direct Processes consist of statements with parameters that provide instructions for initiating such activities as 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.

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 these selection

arguments or list of arguments, allowing you to make common changes or selections where appropriate to the group. 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 will be affected by the command.

Qualifying as Much as Required for Correct Match

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.

Specifying a List

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) will require that PROCA be Process number 16 and PROCB be Process number 17 for the command to apply.

Filling in the Screens

On the screens, you may enter a list of Process names, numbers, and submitters (userid and node ID) to obtain the same results 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 (userid and node) from left to right to construct the list form of this command.

Modifying a Process and Its Status in the TCQ

You can use the CHANGE PROCESS command from the TCQ to:

- ▶ Modify parameters when the Process is in a nonexecuting state.
- ▶ Restart a failed Process.
- ▶ Release a held Process.

Note: Type an R by the Process Name to release a held process through the Operator Table and Selected Process screens.

CHANGE PROCESS Command Format

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

Label	Command	Parameters
(optional)	CH ange PRO CESS	WHERE (PNAME = name (list) PNUMBER = number (list) SUBmitter = (nodename, userid) (list))
		CASE = Yes No
		CLASS = n
		DEST = destination node
		HOLD = Yes No Call
		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])

Required Parameter

WHERE is the only required parameter for the CHANGE PROCESS command. Some of its subparameters may be required.

WHERE (PNAME = name | (list)

PNUMBER = number | (list)

SUBmitter = (nodename, userid) | (list))

specifies which Process(es) to change. Name multiple Processes in the command using the selection criteria method described on page 8-1 if you have similar changes to make to many Processes.

The selection subparameters are optional; however, at least one of PNAME, PNUMBER, or SUBMITTER must be specified. These subparameters are special in that they identify which Processes are selected for the change activity defined by the other parameters.

Optional Selection Subparameters

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.

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 = (nodeid, userid) | (list)

specifies the nodeid and userid 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.

Optional Change Parameters

The following parameters are optional.

CASE = Yes | No

specifies whether parameters associated with accounting data, userid, password, and data set name are case sensitive. The designation refers only to the command, *not to the Process itself*. See page 2-9 for a general overview of case sensitivity.

CLASS = n

determines the node-to-node session on which a Process can execute. Consult your CONNECT:Direct administrator for instructions concerning which class to specify. The range is 1–255.

DEST = destination node

specifies a new destination node. This changes the node that this Process communicates with.

HOLD = Yes | No | Call

specifies whether or not the Process will be placed in the hold queue.

Yes specifies that the Process will remain in the hold queue until one of the following events occurs:

- A CHANGE PROCESS command releases the Process
- A DELETE PROCESS command deletes the Process

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.

No specifies that CONNECT:Direct will not place the Process in the hold queue, but will place it in the WAIT for execution queue (EX). HOLD=NO is the default.

Call specifies that CONNECT:Direct is to place the Process in the hold queue until a VTAM 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 were submitted with HOLD=CALL.

Note: CONNECT:Direct ignores the HOLD parameter if RETAIN=Y.

PRTY = n

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 the priority defined during installation. The range is from 0–15.

RELEASE

specifies the release of the Process for execution from the queue where it is currently residing.

Note: Either specify RELEASE or omit RELEASE—yes and **no** are not valid.

RESTART = [NO |

FIRST = volume sequence number |

FIRST=SER=volume serial number |

LAST = volume sequence number |

LAST = SER = volume serial number |

VOLCNT = n]

specifies the conditions for restarting an interrupted Process.

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.

NO specifies that the copy step should restart at the beginning of the transmission.

FIRST=volume sequence number specifies that the copy step should restart at the beginning of the volume designated by the volume sequence number.

FIRST=SER=volume serial number specifies that the copy step should restart at the beginning of the volume serial given.

LAST=volume sequence number specifies that the copy step should restart at the end of the volume designated by the volume sequence number. LAST specifies that the copy step should restart at the last block on the volume if the output is disk or the last checkpoint on the volume if the output is tape.

LAST=SER=volume serial number specifies that the copy step should restart at the end of the volume serial given. LAST specifies that the copy step should restart at the last block on the volume if the output is disk or the last checkpoint on the volume if the output is tape.

VOLCNT=n specifies that the volume count on the interrupted copy step be changed to the value specified. This can be used to increase the number of output volumes if the copy step was interrupted because the volume count of that step was too small.

RETAIN = Yes | No | Initial

specifies whether or not CONNECT:Direct keeps a copy of the Process in the HOLD queue after the Process has executed.

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.

No specifies that CONNECT:Direct is to delete the process after execution. The default value for RETAIN is NO.

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.

STARTT = ([date | day] [,hh:mm:ssXM])

specifies that the Process not execute until a specified date or time. See page 7-8 for a complete explanation.

On the STARTT parameter in the CHANGE PROCESS command when changing the **date**, **day**, or **hh:mm:ssXM**, you must re-specify even the values that do not change. You cannot specify TODAY or TOMORROW for the **day** subparameter.

Batch Interface Use of CHANGE PROCESS

To issue the CHANGE PROCESS command from the batch interface, perform the following steps:

1. Place the CHANGE PROCESS command in the DMBATCH job stream as described in the *Issuing Commands Through the Batch Interface* chapter beginning on page 3-1.
2. Submit the Process while CONNECT:Direct is running.
3. Verify the results.

You can type **NDM** before the command in the following examples to submit the job using the Command Line Interface (CLI). Refer to the *Issuing Commands Through the Command Line Interface* chapter beginning on page 5-1 for more information on the CLI.

Batch Interface CHANGE PROCESS Example

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 will be deleted after execution.

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

Using CHANGE PROCESS Through the IUI

To change a process in the IUI, perform the following steps:

1. Access the CHANGE PROCESS screen by selecting option **CP** from the Primary Options Menu.

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

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

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

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

DESTINATION NODE  ==> _____
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)
SCHED. START DATE ==> _____ TIME ==> _____
RESTART ALTERATIONS ==> _____
DO YOU WANT THE VALUES OF THIS REQUEST TO BE CASE SENSITIVE? ==>
```

CONNECT:Direct selects the corresponding Process names and numbers and submitters (userid and node) from left to right in constructing the list form of this command. This list fully qualifies your search criteria.

2. Type in at least one of the parameters or a combination of parameters as described in the CHANGE PROCESS command syntax section on page 8-3 or press the **PF1** key for online help.

CONNECT:Direct changes the parameters in the way you indicate for the Processes which match your search criteria.

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

Note: You can also modify certain parameters of a Process from the Selected Process screens by typing a **C** next to the Process Name and making your changes.

Suspending, Flushing, and Deleting Processes

The SUSPEND PROCESS, FLUSH PROCESS, and DELETE PROCESS commands have similar formats and parameters. Their particular functions are as follows:

- ▶ The SUSPEND PROCESS command terminates execution and *puts an executing Process on 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.
- ▶ The FLUSH PROCESS command *terminates an executing Process*. Note that if you specify RETAIN=YES, the flushed Process will remain on 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.
- ▶ The DELETE PROCESS command allows you to specify the name of a *nonexecuting Process* and *remove it from the TCQ*. (Note that the FLUSH PROCESS command stops an *executing Process*.)

If you do not specify the FORCE option for the FLUSH or SUSPEND command, then 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 the FORCE option, then the action taken depends on the STATE and SUBSTATE of the Process for which you issued the FORCE FLUSH or FORCE SUSPEND. You can 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.

When MP=NO is specified, the action taken by FORCE FLUSH or FORCE SUSPEND is as follows:

- ▶ For an LU6.2 Process in a VTAM I/O STATE with a SUBSTATE of 6.2 SEND, RECEIVE, PRERECV, SEND CONFRMD, or SEND ERROR, an APPCCMD with CONTROL=REJECT is issued for the session to reject the conversation.
- ▶ For a non-LU6.2 and non-TCP/IP Process in a VTAM I/O STATE with a SUBSTATE of SEND or RECEIVE, a RESETSR is issued to cancel any pending RECEIVES.
- ▶ For a Process in a SUBTASK or OPEN STATE with no internal CONNECT:Direct LOCKS and no outstanding SDIP requests, a DETACH is issued and the Process ends with a U888 abend.

When MP=YES is specified and the Process that you want to flush or suspend is performing a Run Task step, the following conditions apply for FORCE FLUSH or FORCE SUSPEND:

- ▶ The FORCE FLUSH or FORCE SUSPEND command *must* be entered on the submitting node of the Process.
- ▶ If the Run Task step is executing on the SNODE, a FORCE FLUSH must be issued from the SNODE for the executing Run Task to cleanup sessions and the associated control blocks.
- ▶ FLUSH removes the Process from the TCQ and SUSPEND puts the Process on the HOLD Queue with a status of HO HS.
- ▶ For a non-LU6.2 Process, the Run Task step terminates with an SVTM052I error message, reflecting a message id of SOPD046I for a SUSPEND or SOPD047I for a FLUSH.
- ▶ For an LU6.2 Process, the Run Task step terminates with an SVTM052I error message, reflecting a message id of SVTQ022I for both a FLUSH and a SUSPEND.

SUSPEND PROCESS, FLUSH PROCESS, 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 = (nodename, userid) (list) FORCE†) CASE = Yes No

† The FORCE parameter is not valid for DELETE PROCESS

Required Parameter

WHERE is the only required parameter for the SUSPEND PROCESS, DELETE PROCESS, and FLUSH PROCESS commands. Its subparameters are optional except as described.

WHERE (PNAME = name | (list)
PNUMBER = number | (list)
SUBmitter = (nodeid, userid) | (list)
FORCE)

specifies which Process(es) to suspend, delete, or flush. Name multiple Processes in the command using the search criteria method described on page 8-1.

The subparameters are optional. However at least one of PNAME, PNUMBER, and SUBMITTER must be specified.

Selection 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 to be selected or a list of Process numbers enclosed in parentheses and separated by commas. The range is 1–99999.

SUBmitter = (nodeid, userid) | (list)

specifies the nodeid and userid 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

allows 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.

If FORCE is specified for a Process executing on a LU6.2 session, this parameter may terminate the session immediately, and CONNECT:Direct will not exchange the Process statistics between nodes.

Note: The FORCE parameter is not valid for the DELETE PROCESS command.

Optional Parameter

The CASE parameter is the only optional parameter for the SUSPEND, DELETE, and FLUSH PROCESS commands.

CASE = Yes | No

specifies whether parameters associated with accounting data, userid, password, and data set name are case sensitive. The designation refers only to the command, *not the Process itself*.

Using the Commands Through the Batch Interface

To issue the SUSPEND PROCESS, DELETE PROCESS, or FLUSH PROCESS commands from the batch interface, perform these steps:

1. Place the command in the DMBATCH job stream as described in the *Issuing Commands Through the Batch Interface* chapter beginning on page 3-1.
2. Submit the Process while CONNECT:Direct is running.
3. Verify the results.

You can type **NDM** before the command in the following examples to submit the job using the Command Line Interface (CLI). Refer to the *Issuing Commands Through the Command Line Interface* chapter beginning on page 5-1 for more information on the CLI.

SUSPEND PROCESS Examples

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

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

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

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

DELETE PROCESS Examples

The following command deletes Processes with the number 60584.

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

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

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

FLUSH PROCESS Examples

The following command flushes executing Processes submitted by the user 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))
```

Using the Commands Through the UI

Access the SUSPEND PROCESS, DELETE PROCESS, or FLUSH PROCESS screens in the following manner:

- ▶ Use the Suspend an Executing Process screen to specify the Process that you want to suspend. To get to the screen, select option **PS** from the Primary Options Menu. This screen can also be accessed by entering **SUS** at the Primary Options Menu command prompt.

You can also suspend a Process by typing a **P** next to the Process Name on the Operator Table, the Operator Table - Executing Queue, or the Selected Process screens.

- ▶ Use the Delete A Nonexecuting Process screen to specify the nonexecuting Process that you want to delete. To get to the screen, select option **DP** from the Primary Options Menu.

You can also delete a nonexecuting Process by typing a **D** next to the Process Name on the Operator Table or the Selected Process screens.

- ▶ Use the Flush Process screen to specify the Process that you want to flush. To get to the Flush Process screen, select option **FP** from the Primary Options Menu.

You can also use the Selected Process screen to flush an executing Process by typing an **F** next to the Process Name.

Sample of SUSPEND PROCESS Screen

The following figure illustrates the Suspend an Executing Process screen where CONNECT:Direct builds the SUSPEND PROCESS command.

```
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
```

Note: 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.

CONNECT:Direct selects the corresponding Process names and numbers and submitters from left to right in constructing the list form of this command. This list fully qualifies your search criteria.

Type in at least one of the parameters or a combination of parameters as described in the SUSPEND PROCESS command syntax section on page 8-11 or press the PF1 key for online help. CONNECT:Direct will suspend the Processes which match your search criteria. The case sensitivity designation refers only to the command parameters on the screen, not to the parameters of the Process itself.

Examining Processes in the TCQ

Use the SELECT PROCESS command to examine Processes in the TCQ. You can specify the search criteria and the form in which the information will be 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 associated parameters. Required parameters and keywords are in bold print. Default values for parameters and subparameters are underlined.

Label	Command	Parameters
(optional)	SElect PROCESS	WHERE (<ul style="list-style-type: none"> DEST = node (list) PNAME = name (list) PNumber = number (list) QUEUE = <u>All</u> queue name STATUS = <u>process status</u> (list) SUBmitter = (nodename, userid) (list))
		CASE = Yes No
		FILE PPrint <u>TABLE</u>

Required Parameter

WHERE is the only required parameter for the SELECT PROCESS command. Not all its subparameters are required.

WHERE (DEST = node | (list)

PNAME = name | (list)

PNUMber = number | (list)

QUEUE = All | Exec | Hold | Timer | Wait

STATUS = process status | (list)

SUBmitter = (nodename, userid) | (list))

specifies which Process(es) to select. Name multiple

Processes in the command using the search criteria

method described on page 8-1. The subparameters

PNAME, PNUMber, and SUBmitter are optional, but at

least one must be specified.

Selection Subparameters

DEST = node | (list)

specifies the destination node name of the Process to be selected 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.

PNUMber = number | (list)

specifies the number of the Process to be selected or a list of Process numbers enclosed in parentheses and separating by commas. The range is 1–99999.

QUEUE = All | queue name

specifies Process selection based on the TCQ.

All specifies selection of a Process from all queues. This is the default.

queue name can be one of the following:

Name	Description
E	Executing queue
H	Hold queue

(continued)

Name	Description
T	Timer queue
W	Wait queue

STATUS = process status | (list)

specifies Process selection by status value or a list of status values in parentheses separated by commas.

Values are:

Status Values	Definition
H	All Held Processes
R	All Restart 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
RA	Held for Restart due to Allocation Error
RH	Restart Held
WC	Wait for Connection

SUBmitter = (nodename, userid) | (list)

specifies the nodeid and userid of the user that submitted the Process. Specify a list of SUBmitter IDs by enclosing the IDs in parentheses and separating them by commas.

Optional Parameters

The following are the optional parameters for the SELECT PROCESS command.

CASE = Yes | No

specifies whether parameters associated with accounting data, userid, password, and data set name are case sensitive. The designation refers only to the command, *not the Process itself*. See page 2-9 for a general overview of case sensitivity.

FILE | PPrint | TABLE

specifies the form in which the information will be presented. TABLE is the default. See page 2-11 for detailed information about the output format produced by each of these parameters.

Using SELECT PROCESS Through the Batch Interface

To issue the SELECT PROCESS command from the batch interface, perform the following steps:

1. Place the SELECT PROCESS command in the DMBATCH job stream as described in the *Issuing Commands Through the Batch Interface* chapter beginning on page 3-1.
2. Submit the Process while CONNECT:Direct is running.
3. Verify the results.

You can type **NDM** before the SELECT PROCESS command in the following examples to submit the job using the Command Line Interface (CLI). Refer to the *Issuing Commands Through the Command Line Interface* chapter beginning on page 5-1 for more information on the CLI.

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

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

Using 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 SELECT PROCESS through the IUI, perform the following steps:

1. Access the Select Process screen by selecting option **SP** from the Primary Options Menu. **CONNECT:Direct** builds the **SELECT PROCESS** command from this screen.

```

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

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

PROCESS NUMBERS:
==> _____ ==> _____ ==> _____ ==> _____
PROCESS NAMES:
==> _____ ==> _____ ==> _____ ==> _____
STATUS:          (HO,HR,HI,HE,HC,HP,HS,RH,RA,WC,H,R,W)
==> ___ ==> ___ ==> ___ ==> ___
DESTINATION NODES:
==> _____ ==> _____
==> _____ ==> _____
USER ID:         NODE ID:
==> _____
==> _____
==> _____

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

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

```

2. Type at least one of the parameters or a combination of parameters as described in the **SELECT PROCESS** command syntax section on page 8-16 or press the **PF1** key for online help.

The Processes that match your search criteria are selected for display or print. The case sensitivity designation refers only to the command parameters on the screen, not to the parameters of the Process itself.

3. Select the desired output by typing either **(P)** for print or **(D)** for display on the command line.

The output is presented in tabular format as shown in the following figure. Press the **PF1** key to learn how to browse the display file.

```

      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
      Subnode  => SC.MVS.USERID                   Time   =>
      Other Node => SC.MVS.USERID                 Date   =>
                                                Day    =>
      Userid   => USERID
      Prty     => 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.CNTRL
      Subnode  => SC.MVS.USERID                   Time   =>
      Other Node => SC.MVS.USERID                 Date   =>
                                                Day    =>
      Userid   => USERID

```

In addition to the usual print or display output format, you may select an Operator Table format for your output, as follows:

- ▶ Select the **O** option to access the Operator Table and create a one line summary of each selected Process. See the figure on page 8-21.
- ▶ Select the **S** option to build an Operator Table consisting only of Processes that are currently executing. The table shows how much data was transmitted for a COPY step. Refer to page 8-21 for a description of the display.

Operator Table Summary of Selected Processes

The Operator Table displays the PNODE Processes which satisfy your selection criteria. Obtain the Operator Table display by entering **O** at the command line of the Select Process screen. See page 8-16 for an explanation of the PNAME, PNUMBER, SUBMITTER, USERID, and QUEUE fields.

Note: The Operator Table does not display information about Processes submitted from another node, unless it is in the executing queue.

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 ----- ROW 1 TO 5 OF 5
==> Q          SCROLL ==> PAGE
OPTION  PNAME  PNUMBER  SUBMITTER.NODE  OTHER.NODE  QUEUE
-----
-      DELPROC  100      CD.DALLAS      CD.LOSANGELES  TI HR
      USER01
-      HLDPROC  101      CD.DALLAS      CD.NEWYORK      TI WC
      USER02
-      RELPROC  102      CD.DALLAS      CD.LOSANGELES  HO HI
      USER01
-      SELPROC  105      CD.DALLAS      CD.BOSTON       HO HR
      USER01
-      SUSPROC  104      CD.DALLAS      CD.NEWYORK      EX EX
      USER03

```

From the Operator Table, perform the following functions by typing an option in the OPTION column next to the Process name (PNAME) and pressing the ENTER key:

- ▶ Type **H** to place a nonexecuting Process on the **Hold** queue
- ▶ Type **D** to **delete** a nonexecuting Process from the queue
- ▶ Type **R** to **release** a held Process
- ▶ Type **P** to **suspend** a Process from the executing queue
- ▶ Type **S** to **select** a Process for detailed display

Note: When you release a Process which contains the RETAIN=YES parameter, CONNECT:Direct copies the Process and assigns a new Process number.

If the CONNECT:Direct 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 the ENTER key to refresh the screen.

To view the message text, enter END to leave the Operator Table and return to the Select Process screen. Type M on the command line, and press ENTER.

Accessing Execution Queue Status on the Operator Table

The Operator Table/Executing Queue display consists only of Processes that are *currently executing*. Obtain the display by entering S at the command line of the Select Process screen. The Operator Table/

Executing Queue screen shows how much data was transmitted by the COPY step including:

- ▶ The number of blocks for block-mode transmissions
- ▶ The number of records for record-mode transmissions
- ▶ The number of RUs (request/response units)
- ▶ The number of I/O bytes
- ▶ The number of VTAM bytes
- ▶ The compression factor

The following figure is a sample of the display as it appears while two Processes are executing.

```

----- OPERATOR TABLE / EXECUTING QUEUE ----- ROW 1 TO 2 OF 2
==> Q          SCROLL ==> PAGE
OPTION  PNAME  PNUMBER  SUBMITTER.NODE  OTHER.NODE  QUEUE
-----
-      HLDPROC  101      CD.DALLAS  CD.NEWYORK  EX EX
      USER01
      Blks      => 59      Recs => 2,298      RUS => 45
      I/O Bytes => 184,080      Compression
      VTAM bytes => 184,320      Factor => 0.1%

-      SUSPROC  104      CD.DALLAS  CD.NEWYORK  EX EX
      USER02
      Blks      => 75      Recs => 3,112      RUS => 55
      I/O Bytes => 227,760      Compression
      VTAM bytes => 225,280      Factor => 1.1%

```

Indicate the options **S** (Select Process) or **P** (Suspend Process) in the OPTION column next to the Process name to select or suspend a Process.

Press **ENTER** to update the display and monitor execution.

Viewing and Controlling a Process Through the SELECTED PROCESS Screen

Obtain the Selected Process screen by typing **S** in the OPTION column next to the name of the Process to be selected in the Operator Table or the Operator Table/Executing Queue screen.

The screen 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 =>          Submitter => submitterid1
Userid => userid1
Process Number =>          Other Node=> node.name1      Queue =>
Queueing Prty =>          Class =>                    Status =>
Scheduled Time =>          Date =>                    Day =>
Retain Status =>          sub state =>                State =>
Step Name =>              Function =>                Member =>
Blks =>                   Recs =>                    RUS =>
Submitted Class=>          Max Class =>                Sess.ID=>
Last Msgid =>              Last RC =>                RetProc=>
SENDING SIDE              COMMID ==>
Session restrt =>          Dyn rstrt =>                RouteID=> routeid1
Volume seq no. =>          volser =>                    TTRN =>
Sending Dsname =>
Receiving DSN =>
I/O bytes =>
VTAM bytes =>
Compression Factor =>

```

From the Selected Process screen, you can issue the following Process control commands by typing **F**, **P**, **D**, **H**, **R**, or **C** at the command line and pressing the **ENTER** key:

- ▶ Type **F** to flush the currently executing Process
- ▶ Type **P** to suspend the currently executing Process
- ▶ Type **D** to delete a nonexecuting Process
- ▶ Type **H** to put a nonexecuting Process in the HOLD queue
- ▶ Type **R** to release a nonexecuting Process
- ▶ Press **ENTER** to monitor the currently executing Process execution
- ▶ Type **C** to change Other Node, Queueing Priority, Scheduled Time, Scheduled Day, and Retain Status in a nonexecuting Process

To change the parameters move the cursor to the appropriate field on the screen and make the change. Press the **ENTER** key after you have made all your changes.

If the Process completes execution while the Selected Process screen is displayed, pressing the **ENTER** key returns you to the Operator Table screen.

Examining Process Results and Statistics

This chapter contains information about how to use the `SELECT STATISTICS` command to retrieve statistics records based on various criteria and how to display the results. Use this command to see the results of a Process that has finished executing.

Statistics Log Records

CONNECT:Direct keeps a record of operations and activity in a Statistics Log file made up of records similar to the following table:

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 MVS internal reader (JES)
Run Task	A program attached to the Process as a subtask
Submit within Process	A Process that was submitted within another Process

Refer to the *Using CONNECT:Direct Exits* chapter in the *CONNECT:Direct for VM/ESA Administration Guide* for a complete table of statistics record types.

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
 - Userid
 - 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 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 follows:

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

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

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

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

- ▶ In the Submit within Process record, the Submit Node indicates which node submitted the Process.

Elapsed Time Significance

Interpret the elapsed time between the start and stop times as follows for the different record types:

- ▶ For Copy Termination, the elapsed time includes file allocation on both nodes as well as 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.

Retrieving Statistics Records

Use the SELECT STATISTICS command to retrieve and examine statistic log records. You can select records based on certain conditions and can indicate if you want the output displayed, printed, or saved in a file for later processing.

See information about the FILE subparameter in the *Indicating Output Destination* section beginning on page 2-11 for how to save your output in a file for later processing.

SELECT STATISTICS Command Format

The SELECT STATISTICS command has the following format and associated parameters. 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)
		PNAME = name (list)
		PNUMBER = number (list)
		STARTT = ([date day] [,hh:mm:ssXM])
		STOPT = ([date day] [,hh:mm:ssXM])
		USER = name (list)
		SNODE = name (list)
		TYPE = ID (list)
		FNAME = dsname (list) *
		CASE = YES NO
		ARCHDSN = dsname (list)
		FILE PRINT <u>TABLE</u> SUMMARY

Required Parameter

WHERE is a required parameter for the SELECT STATISTICS command.

WHERE (CCODE = (condition, completion code)

**EXCLUDE = (MEMber | MCR) | (WTO) | (NOTWTO) |
(list)**

PNAME = name | (list)

PNUMBER = number | (list)

STARTT = ([date | day] [,hh:mm:ssXM])

STOPT = ([date | day] [,hh:mm:ssXM])

USER = name | (list)

SNODE = name | (list)

TYPE = ID | (list) | *

FNAME = dsname | (list)

CASE = YES | NO

ARCHDSN = dsname | (list))

specifies which statistics record(s) you want to examine. The subparameters, such as PNAME, PNUMBER, and STARTT, are optional, but at least one must be specified.

Selection Subparameters

CCODE = (condition, completion code)

specifies selection by completion code.

condition specifies a relationship to the completion code given in the subsequent positional parameter. The options for specifying condition are:

GT 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

completion specifies a completion code value ranging from 1 to 2,147,483,647 to allow for all values that can be passed by the RUN TASK statement. This last value represents a maximum 31-bit binary number.

For example, if CCODE = (GT,0) is specified, you will see statistics records in which the step completion code is greater than zero, as long as the records also meet other specified criteria.

EXCLUDE = (MEMBer | MCR) | (WTO) | (NOTWTO) | (list)

excludes 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.

MEMBer | MCR specifies whether or not to exclude the PDS member copy record for PDS copies on the statistics report.

WTO specifies that CONNECT:Direct write-to-operator (WTO) messages be excluded from the SELECT STATISTICS command. Dynamic allocation messages are represented as WTOs in the statistics file.

NOTWTO specifies that only WTO records be displayed (exclude everything that is NOT a WTO record).

PNAME = name | (list)

specifies selection by Process name. A list of Processes can be specified 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.

PNUMBER = number | (list)

specifies selection by Process number. A list of Processes can be specified by enclosing them in parentheses. The range is 1-99999.

STARTT = ([date | day] [,hh:mm:ssXM])

specifies that CONNECT:Direct search for statistics records by a designated starting date and time. The DATE, DAY, and TIME are positional parameters. If you do not specify the DATE or DAY, a comma must precede the time.

date specifies the date from which the statistics records are selected. CONNECT:Direct is Year 2000 compliant. For detailed information on Year 2000 compliance, refer to the *CONNECT:Direct for VM/ESA Release Notes*.

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 date values.

Note: You may omit the separators only for transfers between mainframe nodes. Sterling Commerce recommends the use of separators to guarantee transfers between all platforms.

After you designate the date order in your initialization parameters, you can use the following date formats:

If you specify DATEFORM=MDY, CONNECT:Direct interprets the date format as:

- ▶ mm/dd/yy or mm/dd/yyyy
- ▶ mm.dd.yy or mm.dd.yyyy

If you specify DATEFORM=DMY, CONNECT:Direct interprets the date format as:

- ▶ dd/mm/yy *or* dd/mm/yyyy
- ▶ dd.mm.yy *or* dd.mm.yyyy

If you specify DATEFORM=YMD, CONNECT:Direct interprets the date format as:

- ▶ yy/mm/dd *or* yyyy/mm/dd
- ▶ yy.mm.dd *or* yyyy.mm.dd

If you specify DATEFORM=YDM, CONNECT:Direct interprets the date format as:

- ▶ yy/dd/mm *or* yyyy/dd/mm
- ▶ yy.dd.mm *or* yyyy.dd.mm

CONNECT:Direct processes Julian dates the same as previous releases. The only change in Julian date processing is the provision for four-digit years. The following formats are valid:

- ▶ yyddd *or* yyyyddd
- ▶ yy/ddd *or* yyyy/ddd
- ▶ yy.ddd *or* yyyy.ddd

If only date is specified, the time defaults to 00:00.

day specifies the day of the week to be selected. 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.

hh:mm:ssXM indicates the time of day in hours (hh), minutes (mm), and seconds (ss) to be selected. XM can be set to AM or PM. You can express the time of day using the 24-hour clock or the 12-hour clock. If you use the 24-hour clock, valid times are 00:00–24:00. If you use the 12-hour clock, 1:00 hours could be expressed as 1:00AM, and 13:00 hours could be expressed as 1PM.

If you use neither AM or PM, CONNECT:Direct assumes the 24-hour clock. You need not specify minutes and seconds. You can also specify NOON, which will search for the statistics records at noon, or MIDNIGHT, which will search for the statistics records at midnight. The default for the time is 00:00:00, the beginning of the day.

If you specify time of day but not date, the output will show the first available entry in the statistics log for that

time of day. Statistics from all later times and dates will be output up to and including the stop time.

If you do not specify the STARTT parameter, the search criterion will consist of any time before the STOPT parameter.

STOPT = ([date | day] [,hh:mm:ssXM])

specifies that CONNECT:Direct search statistics records for up to and including the designated DATE, DAY and TIME positional parameters. If you do not specify the DATE or DAY, you must precede the time with a comma.

date specifies the date from which the statistics records are selected. CONNECT:Direct is Year 2000 compliant. For detailed information on Year 2000 compliance, refer to the *CONNECT:Direct for VM/ESA Release Notes*.

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 date values.

Note: You may omit the separators only for transfers between mainframe nodes. Sterling Commerce recommends the use of separators to guarantee transfers between all platforms.

After you designate the date order in your initialization parameters, you can use the following date formats:

If you specify DATEFORM=MDY, CONNECT:Direct interprets the date format as:

- ▶ mm/dd/yy or mm/dd/yyyy
- ▶ mm.dd.yy or mm.dd.yyyy

If you specify DATEFORM=DMY, CONNECT:Direct interprets the date format as:

- ▶ dd/mm/yy or dd/mm/yyyy
- ▶ dd.mm.yy or dd.mm.yyyy

If you specify DATEFORM=YMD, CONNECT:Direct interprets the date format as:

- ▶ yy/mm/dd or yyyy/mm/dd
- ▶ yy.mm.dd or yyyy.mm.dd

If you specify DATEFORM=YDM, CONNECT:Direct interprets the date format as:

- ▶ yy/dd/mm or yyyy/dd/mm
- ▶ yy.dd.mm or yyyy.dd.mm

CONNECT:Direct processes Julian dates the same as previous releases. The only change in Julian date processing is the provision for four-digit years. The following formats are valid:

- ▶ yyddd or yyyyddd
- ▶ yy/ddd or yyyy/ddd
- ▶ yy.ddd or yyyy.ddd

If only date is specified, the time defaults to 00:00.

day specifies the day of the week to be selected. Valid values include MOnday, TUEsday, WEDnesday, THursday, FRiday, SATurday, and SUnDay. You can also specify TODAY.

hh:mm:ssXM indicates the time of day in hours, minutes, and seconds to be selected. Set XM to AM or PM. Express the time of day using the 24-hour clock or the 12-hour clock. If the 24-hour clock is used, valid times are 00:00–24:00. If the 12-hour clock is used, 1:00 hours would be expressed as 1:00AM, and 13:00 hours would be expressed as 1PM. If AM and PM are not used, the 24 hour clock is assumed.

You can also specify NOON, which indicates that records ending at noon will be searched for, or MIDNIGHT, which specifies that records ending at midnight will be searched for.

The default for the time is 24:00:00, the end of the day.

If you specify time of day but not date, all statistics through the most recent entry for that time of day is output.

If you do not specify the STOPT parameter, CONNECT:Direct selects statistics through the present time.

USER = name | (list)

limits the selected statistics records to those that were 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.

SNODE = name | (list)

limits the selected statistics records to those that were 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 **SYS\$** in the first four characters of the **USER** 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.

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 *overrides the EXCLUDE parameter*. When **TYPE** is not specified, the record types selected are determined by the output option chosen, and can be affected by the **EXCLUDE** subparameter. When the **PRINT** or **TABLE** output option is specified, 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.

The following table describes the valid ID types.

ID	Description
PS	Process Submit
CT	Copy Termination
MC	PDS Member Copy
RT	Run Task
RJ	Run Job
SW	Submit Within a Process
WO	Write to Operator
PT	Process Termination

See the *Using CONNECT:Direct Exits* chapter in the *CONNECT:Direct for VM/ESA Administration Guide* for a list of all statistics record type identifiers.

FNAME=dsname | (list)

limits the selected statistics records to those that contain the specified filename. The FNAME subparameter is meaningful for the following record types:

- ▶ CT Copy Termination
- ▶ RJ Run Job
- ▶ SD Start CONNECT:Direct
- ▶ SW Submit within Process

The meaning of the filename within these records is unique for each record type. For example, the Run Job record contains the filename of the submitted JCL. Filenames can be up to 64 characters in length and can contain lowercase characters.

CASE=YES | NO

specifies whether lowercase or mixed-case data is permitted for the USER, SNODE, and FNAME subparameters. The CASE subparameter overrides the global CASE option defined at signon for the purpose of the SELECT STATISTICS command.

YES folds the data in USER, SNODE, and FNAME to uppercase regardless of the actual data specified.

NO preserves the actual case entered for the USER, SNODE, and FNAME subparameters.

The CASE defaults to the setting defined within the session defaults if nothing is specified.

ARCHDSN=dsname | (list)

specifies that CONNECT:Direct is to search the archived statistics files named, *instead of the statistics log currently being used by the DTF*. Archived statistics files are used to 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.

Optional Parameters

The following are the optional parameters for the SELECT STATISTICS command.

FILE | PRint | TABLE | SUMMARY

specifies the output format. Indicate only one. TABLE is the default.

See page 2-11 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 that shown on page 9-17.

Using SELECT STATISTICS Through the Batch Interface

To issue the SELECT STATISTICS command from the batch interface, perform the following steps:

1. Place the SELECT STATISTICS command in the DMBATCH job stream as described in the *Issuing Commands Through the Batch Interface* chapter beginning on page 3-1.
2. Submit the Process while CONNECT:Direct is running.
3. Verify the results.

You can type **NDM** before the SELECT STATISTICS command in the following examples to submit the job using the Command Line Interface (CLI). Refer to the *Issuing Commands Through the Command Line Interface* chapter beginning on page 5-1 for more information on the CLI.

Examples

In this example, you define a search for statistics records based on start and stop date and time, but exclude write-to-operator messages.

```
SEL STAT WHERE ( -  
                EXCLUDE=(WTO) -  
                STARTT=(11/24/96,06:45:00) -  
                STOPT=(11/25/96,8AM) -  
                )
```

In this example, you define a search for statistics records based on start and stop date and time, excluding write-to-operator and PDS member copy statistics records.

```
SEL STAT WHERE ( -  
                EXCLUDE=(WTO,MEMB) -  
                STARTT=(11/21/96,15:25:00) -  
                STOPT=(11/25/96,15:30:00) -  
                )
```

This search for statistics records is based on process name, start and stop date and time, excluding write-to-operator statistics messages.

```
SEL STAT WHERE ( -  
                EXCLUDE=(WTO) -  
                PNAME=BILLING -  
                STARTT=(12/28/96,06:00) -  
                STOPT=(12/28/96,23:00) -  
                )
```

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
```

This command searches for Process submission records for Processes in which the SNODE during the month of November of 1996 is any node in Dallas. 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. The output prints out.

```

SEL STAT WHERE ( -
                TYPE=( PS, SW) -
                SNODE=DALLAS* -
                ARCHDSN=SYS.ARCH.STATS.M9611 -
                ) PRINT

```

The preceding command results in the following report.

RESTON.PROD		CONNECT:Direct SELECT STATISTICS	DATE: 11/01/1996
-			
Function	=>PROCESS SUBMIT	Start Time	=>08:37:12
Process Name	=>TESTPROC	Stop Time	=>08:37:14
Process Num	=>17	Comp Code	=>0000000
		Comp Msg	=>SSPA001I
Userid	=>JONES		
Secondary	=>DALLAS.TEST		
-			
Function	=>SUBMIT W/IN PROC	Start Time	=>09:53:25
Process Name	=>RLTPROCL	Stop Time	=>09:53:26
Process Num	=>79	Comp Code	=>0000000
		Comp Msg	=>SSUB000I
Userid	=>SMITH		
Secondary	=>DALLAS.DEVELOP		
-			
Function	=>PROCESS SUBMIT	Start Time	=>12:15:01
Process Name	=>TESTPROC	Stop Time	=>12:15:03
Process Num	=>79	Comp Code	=>0000000
		Comp Msg	=>SSPA001I
Userid	=>BUCHAN		
Secondary	=>DALLAS02		

The following command searches for signon and signoff records that match the date for today and have the userid SEAN.

```

SEL STAT WHERE ( -
                TYPE=( SI, SO) -
                USER=SEAN -
                STARTT=( TODAY) -
                STOPT=( TODAY, 24:00:00) -
                ) TABLE

```

The preceding command results in the following report:

CD.MVS.USER1		CONNECT:Direct SELECT STATISTICS		DATE: 11/01/1996
Function	=>SIGN ON	Time	=>10:05:44	
Userid	=>SEAN	Comp Code	=>0000000	
Node	=>CD.MVS.USER1	Comp Msg	=>SAFA000I	
Function	=>SIGN OFF	Time	=>10:05:44	
Userid	=>SEAN	Comp Code	=>0000000	
Node	=>CD.MVS.USER1	Comp Msg	=>SAFA000I	
Function	=>SIGN ON	Time	=>14:05:44	
Userid	=>SEAN	Comp Code	=>0000000	
Node	=>CD.MVS.USER1	Comp Msg	=>SAFA000I	
Function	=>SIGN OFF	Time	=>17:28:00	
Userid	=>SEAN	Comp Code	=>0000000	
Node	=>CD.MVS.USER1	Comp Msg	=>SAFA000I	

Sample Output

The following figure shows a sample output from a SELECT STATISTICS command.

CD.NYC		CONNECT:Direct SELECT STATISTICS		DATE: 11/25/1996
Function	=>PROCESS SUBMIT	Start Time	=>06:01:58	
Process Name	=>PAYROLL	Stop Time	=>06:01:58	
Process Num	=>12	Comp Code	=>0000000	
		Comp Msg	=>SSPA001I	
Userid	=>JONES	Step Name	=>	
Secondary	=>			
Function	=>MEMBER	Start Time	=>06:02:03	
Process Name	=>PAYROLL	Stop Time	=>06:02:06	
Process Num	=>12	Comp Code	=>0000000	
		Comp Msg	=>SCPA000I	
Userid	=>JONES	Step Name	=>STEP1	
Primary	=>CD.DALLAS			
Name:	From =>XYZ	To	=>ABC	
	recs =>0	blks	=>113	
Function	=>COPY	Start Time	=>06:01:58	
Process Name	=>PAYROLL	Stop Time	=>06:02:07	
Process Num	=>12	Comp Code	=>0000000	
		Comp Msg	=>SCPA000I	
Userid	=>JONES	Step Name	=>STEP1	
Primary	=>CD.DALLAS			
From (Pnode				
Dsn=TECHRPT.FILE)				
recs =>0	blks		=>113	
I/O Bytes =>351,360				
VTAM Bytes =>351,586				
Cmpr Perc =>-0.1%				
VOL=SER =>WK0004				
To (Snode				
Dsn=NEWREPT.FILE				
recs =>0	blks		=>113	
I/O Bytes =>351,360				
VTAM Bytes =>351,586				
Cmpr Perc =>-0.1%				
VOL=SER =>WK0007				
Function	=>PROCESS DURATION	Start Time	=>06:01:58	
Process Name	=>PAYROLL	Stop Time	=>06:02:07	
Process Num	=>12	Comp Code	=>0000000	
		Comp Msg	=>SVTM100I	
Userid	=>JONES			
Secondary	=>CD.DALLAS			

Using SELECT STATISTICS Through the UI

Use SELECT STATISTICS to select, view, and print CONNECT:Direct statistics information. Type **SS** from the Primary Options Menu to access the SELECT STATISTICS screen.

```
node.name          SELECT STATISTICS
CMD ==>

                                hh:mm
                                yyyy/mm/dd
                                yyyy.ddd
                                CMD: DIS

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

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

START DATE ==> _____
START TIME ==> _____ (HH:MM:SSXM)
STOP DATE  ==> _____
STOP TIME  ==> _____ (HH:MM:SSXM)

CONDITION CODE: ==> _____

EXCLUDE ( MEMBER RECS ==> Y   WTO RECS ==> Y   ^WTO RECS ==> N )

CHANGE EXTENDED OPTS: ==> N

CMD: S ... SUMMARY TABLE  D ... DISPLAY REPORT  P ... PRINT REPORT
```

On the command line type the output format as follows:

- ▶ Type **D** to display the output on the screen.
- ▶ Type **P** to send the output to the printer.
- ▶ Type **FIL** to view statistic records in an unformatted form.
- ▶ Type **S** to display a 3-line summary per statistics record on the screen. See the Statistics Summary figure on page 9-17.

To access the SELECT STATISTICS EXTENDED OPTIONS screen, type **Y** in the CHANGE EXTENDED OPTS field.

```

node.name                                SELECT STATISTICS
                                         EXTENDED OPTIONS
                                         hh:mm
                                         yyyy/mm/dd
                                         yyyy.ddd

RECORD      ==>  __      ==>  __      ==>  __
TYPES:      ==>  __      ==>  __      ==>  __

SNODE NAME: ==>  _____

USER ID:    ==>  _____

FILENAME:   ==>  _____

SEARCH      ==>  _____
ARCHIVED    ==>  _____
DATASETS:   ==>  _____

DO YOU WANT THE VALUES OF THIS REQUEST TO BE CASE SENSITIVE? ==> N

```

See the description of valid parameters for the SELECT STATISTICS command on page 9-4 for information on how to complete the fields that display or press **PF1** for online help.

Summarize Process Statistics

To show a summary of Process statistics, type **S** in the CMD field from the Select Statistics screen and press **ENTER**. The following Statistics Summary screen displays.

```

-----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
-----
*  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 userid. The third line of each entry is the message ID, the return code, and the ending date and time of that Process.

Note: An asterisk (*) before the Process name indicates a non-zero return-code.

From the STATISTICS SUMMARY screen, you can perform the following tasks:

- ▶ 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 shown in this figure.

```

node.name          COPY                               hh:mm
CMD ==>
Copy operation successful.

Process Name      : ONESTEP      Submitter : SC.MVS.USERID
Userid           : TBUSERID
Process Number   : 3             Other Node: SC.MVS.JCARR1   P Snode : SNODE
Starting Time    : 15:13:57      Date       : 12/02/96      Sending Node : SNODE
Ending Time      : 15:14:02      Date       : 12/02/96      RC       : 00000000
Step Name        : STEP1         Rusize    : 1,024         MSGID    : SCPA000I
                  *COMPRESSED*   *CKPTED*

                          SENDING STATS
Dataset Name      : USERID.TEMP.JCL
Block count       : 1           Records    : 0           RUS      : 1
I/O Bytes        : 800          VTAM bytes: 274
Members          : 0           Aliases    : 0           Comp%    65.8%
Members Sent     : 0           Aliases Sent : 0

                          Receiving STATS
Dataset Name      : USERID.TEMP.OUT
Block count       : 1           Records    : 0           RUS      : 1
I/O Bytes        : 800          VTAM bytes: 274
Members          : 0           Aliases    : 0           Comp%    65.8%
1st 8 volumes    : M80011

```

Display Statistics

To obtain the Display option, type **D** in the CMD field from the SELECT STATISTICS screen and press **ENTER**.

The statistics display function supports 133 character records and horizontal scrolling using the standard ISPF scrolling function keys.

The output format resembles the figure on page 9-15.

```

BROWSE --- SYS94341.T120948.RA000.JCARR1.NDMAPI ----- LINE 00000000 COL 001
080
COMMAND ==>>>                                     SCROLL ==>>> PAGE
***** TOP OF DATA *****
=====
QB.MVS.V20200          SELECT STATISTICS          DATE : 12/07/1996
=====

-----
Function      => COPY                Start Time => 10:45:31
Process Name => ATT16                Stop Time  => 10:45:36
Process Num  => 906                  Comp Code  => 00000000
                                           Comp Msg   => SCEPA000I
Userid       => MFRAN1
Secondary    => QD.MVS.V10702        Step Name  => STEP1

**** CHECKPOINTED
From ( Pnode
Dsn=CSGQAL.IN.STRESS16)
      recs => 250                    blks => 1
      I/O BYTES => 20,000
      VTAM BYTES => 16,388
      Cmpr Perc => 18.1%
      VOL=SER=> USER12
To   ( Snode

```

Output to the system console and messages in response to console commands is formatted as shown in the following figure.

```

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
                                           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(SUBADD0
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.STOURL.SYSIN.T632272
=====
USER01.MVS60.N1          SELECT STATISTICS          DATE : 10/24/1996
=====

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

```

Viewing the Network Map

Use the information in this chapter to retrieve records from the Network Map file or translate TCP/IP host names to network addresses. For more information about the CONNECT:Direct Network Map, see the *CONNECT:Direct for VM/ESA Administration Guide*.

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. Your system administrator maintains the Network Map file.

SELECT NETMAP Command Format

The SELECT NETMAP command uses the following format and associated parameters. The required parameters are in bold print. Default values for parameters and subparameters are underlined.

Label	Command	Parameters
(optional)	SElect NETMAP	WHERE (NODE = (<u>node</u> <u>generic</u> (<u>list</u>)))
		<u>Print</u> <u>DISplay</u>

Required Parameters

WHERE is the only required parameter for the SELECT NETMAP command.

WHERE (NODE = (node | generic | (list))

specifies which Network Map node definition(s) you want to examine. NODE is the only subparameter of the WHERE parameter.

NODE=(node | generic | (list))

specifies the Network Map node name(s) that will be selected. Specify a list of node names by enclosing them in parentheses and separating each by a comma or a blank.

node specifies a node name. The node is a 1–16 character alphanumeric string, with the first character alphabetic.

generic specifies a generic selection of nodename(s). 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*** would provide information about the SANDIEGO and SANFRAN nodes. If you enter only an asterisk (*), CONNECT:Direct will display (or print) all nodes that you are authorized to use.

Optional Parameters

The following are the optional parameters for the SELECT NETMAP command.

PRint | DISplay

indicates the output format. Indicate only one. See page 2-11 for a complete description of the output format produced by these parameters.

Batch Interface Use of SELECT NETMAP Command

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.

Example

The following command displays all Network Map entries for node names that begin with *SAN*.

```
SIGNON
SEL NET WHERE (NODE=(SAN*))
SIGNOFF
```

The report in the following figure is produced if the only node name in the Network Map which begins with *SAN* is the *SANFRAN* node.

```
=====
                        SELECT NETWORK MAP
=====
Node Name   : SANFRAN           VTAM Applid  : CDAPPL1
Max Parsess : 8                 Def Ses Class : 2
Session Type: SNA               Environment   :
Node Status : Internal, Send
Network id  : NETSAN
```

IUI Use of SELECT NETMAP Command

Follow these steps to use the SELECT NETMAP OR TCP INFORMATION screen to select a Network map record in the CONNECT:Direct IUI.

1. Select option **NM** from the Primary Options Menu to access the screen.

```
node.name          SELECT NETMAP OR TCP INFORMATION          hh:mm
CMD ==>

NETMAP INFORMATION

NODE KEY(S): ==> _____ ==> _____
              ==> _____ ==> _____
              ==> _____ ==> _____
              ==> _____ ==> _____

OUTPUT DESTINATION ==> DIS (DIS-DISPLAY,PR-PRINT)
```

2. To select a record, enter the selection criteria for the node(s) you want to examine and press **ENTER**.

Note: You can request wildcard searches by using an asterisk. For example, entering an asterisk (*) in the first node key field selects all nodes and **D*** selects all nodes that begin with **D**.

See the description of valid SELECT NETMAP parameters beginning on page 10-2 or use the online help. The output display is the same as that shown on page 10-3.

The TCP information fields are for creation of the SELECT TCPXLAT command described in the next section and should not be filled in for the SELECT NETMAP command.

Using the Shared File System

This chapter provides information on the components of the Shared File System(SFS) and a brief description of how to request a SFS file.

Understanding the Shared File System

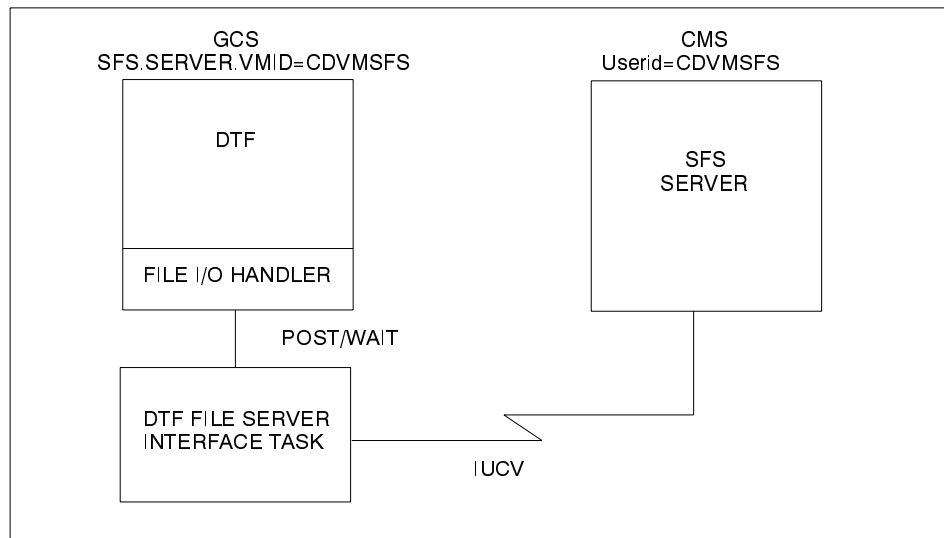
This section introduces the SFS components and describes how these components use the CMS multitasking feature to allow multiple concurrent user access to SFS data.

Shared File System Components

The internal components of the SFS include:

- ▶ DTF File I/O Handler
- ▶ DTF File Server Interface Task
- ▶ CMS File Request Server

This figure illustrates the SFS internal components.



Refer to the *Tuning CONNECT:Direct for VM/ESA* chapter in the *CONNECT:Direct for VM/ESA Installation Guide* for guidelines on how to tune the SFS Server after your installation.

Understanding How a Shared File System Request Works

If you specify the SFS.SERVER.VMID=userid at the DTF initialization time, the DTF AUTOLOGs the named SFS Server, if it is not already logged on. At the end of initialization, the DTF attempts to contact the SFS Server. Once the connection completes, you will see a connection complete message which notifies you that you can access the SFS files.

Use the SFSDIR keyword in the COPY statement to specify the Processed file is an SFS file. The SFSDIR keyword specifies the SFS file-pool-id and directory which contains the file.

Note: SFS files can be the source file, target file, or both.

Refer to the *CONNECT:Direct Process Guide* for more information and sample Processes using SFS on the COPY statement.

Overview of a Shared File System Request

Use the previous figure and the following steps to see an overview of an SFS request.

1. The DTF File I/O Handler receives a request for an SFS file.
2. The DTF File I/O Handler creates a control block with the information about the requested SFS file and the userid of the submitter of the Process.

3. The control block remains in a wait queue for processing by the DTF File Server Interface Task.
4. The DTF File I/O Handler posts the DTF File Service Interface Task to indicate a request for an SFS file.

Note: The File I/O Handler issuing the request waits until notification is received that the I/O request is complete.

5. The File Service Interface Task sends the requested control block information to the CMS File Request Server through the IUCV connection.
6. The CMS File Request Server services the request and returns the control block to the File Service Interface Task to post the request as complete in the DTF File Interface Handler.

CONNECT:Direct DTF Commands for SFS File Interface Task

At the SOIA000A WTOR, which is the operator interface WTOR, you can issue the following commands to the SFS File Interface Task:

R nn SFS INIT

initializes the SFS File Interface Task.

R nn SFS STOP | SHUT | SHUTDOWN Q | QUIESCE | IMMEDIATE | I
quiesces or immediately stops the SFS File Interface Task.

R nn SFS STATUS | STAT L | LONG | S | SHORT

displays the current status of the SFS File Interface Task and its control blocks. The LONG display shows additional information about the interrupt trace table and the current processes that are using the SFS Server. This information is normally used for debugging purposes.

R nn SFS TRACE [ON | OFF]

shows the current trace status for the SFS File Interface Task (if no operand is included), or turns on/off the tracing.

Tracing is used for debugging purposes and should only be turned on when requested to do so by Sterling Commerce Customer Support.

SFS Server Operator Commands

The following commands can be issued from the virtual machine console of the SFS Server:

TRace ON | OFF | PERF

shows the current status of the SFS Server tracing (if no operand is included), or turns on/off the tracing. TRACE OFF turns off both the detail tracing (written to FILEDEF SYSOUT) and the performance tracing (written to FILEDEF PERF). To turn on only the performance tracing, specify TRACE OFF, followed by TRACE PERF. Tracing is normally used for debugging purposes and should only be turned on when asked to do so by Sterling Commerce Customer Support.

STatus or STate

shows the current status of the connected DTFs and the current file threads that are active. The command is normally used for debugging purposes.

STOP

shuts down the SFS Server immediately and severs all IUCV connections.

See the *CONNECT:Direct Process Guide* for examples of Processes using the Shared File System.

Using the Utility Programs

CONNECT:Direct provides several utility programs 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
- ▶ Convert files using DM RTPOMV (MVS to VM) and DM RTPOVM (VM to MVS)
- ▶ Notify any userid on the host machine through a punched file DMGPUN01

User Notification Programs (DMNOTIFY and DMNOTFY2)

CONNECT:Direct provides sample programs that notify the console operator or IUI users regarding the success or failure of any Process step. The sample programs are supplied in VM/ESA load library.

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 two parameters passed to it as follows.

CL4'GOOD' | CL4'FAIL'

(first positional parameter) tells the program the correct message to output. Code the parameter as shown, including the single quotes.

file name

(second positional parameter) is the name of the file that is being copied.

The example in the following figure demonstrates a CONNECT:Direct for VM/ESA Process using the DMNOTIFY program.

```
COPY1  PROCESS PNODE=CDA SNODE=CDB
STEP01 COPY -
        FROM ( -
                PNODE -
                DSN='FILE1 JCL' -
                LINK=(SMITH,BOB,RR,125) -
                DISP=SHR -
                ) -
        TO ( -
                SNODE -
                DSN='FILE2 JCL' -
                DISP=SHR -
                )
STEP02 IF (STEP01=0) THEN
NOTIFYG RUN TASK (PGM=DMNOTIFY, -
                  PARM=(CL4'GOOD', 'FILE1 JCL')) PNODE
ELSE
NOTIFYF RUN TASK (PGM=DMNOTIFY, -
                  PARM=(CL4'FAIL', 'FILE1 JCL')) PNODE
EIF
```

The DMNOTIFY program is attached on the PNODE. A parameter list containing the GOOD or FAIL message criteria and the 'FILE1 JCL' file name is passed to DMNOTIFY. If the COPY statement is successful, the first RUN TASK step executes and you see the following message.

```
+++++++ CONNECT:Direct FILE TRANSFER WORKED ++++++
+++++++ CONNECT:Direct FILE TRANSFER WORKED ++++++

        PNODE=CDA
        SNODE=CDB
REQUESTOR=userid
        DSN =FILE1 JCL

+++++++ CONNECT:Direct FILE TRANSFER WORKED ++++++
+++++++ CONNECT:Direct FILE TRANSFER WORKED ++++++
```

If the COPY statement is unsuccessful, the second RUN TASK step executes and you see the following messages.

```
***** CONNECT:Direct FILE TRANSFER FAILED *****
***** CONNECT:Direct FILE TRANSFER FAILED *****

      PNODE=CDA
      SNODE=CDB
      REQUESTOR=userid
      DSN =FILE1 JCL

***** 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 CONNECT:Direct users of various conditions. The program accepts a variable length parameter list that is used to broadcast notification messages to users, as follows:

CL4'GOOD' | CL4'FAIL'

(first positional parameter) indicates the success or failure of the Process step. Code the parameter as shown including the single quotes.

file name

(second positional parameter) is the name of the file that is being copied.

userid | userid list

(last parameter or list of parameters) contains the VM/ESA userid(s) to be notified. Use a comma to separate the userids.

The following RUN TASK statement parameters are acceptable for interfacing with this program:

```
RUN TASK (PGM = DMNOTFY2
          PARM = (CL4'GOOD' | CL4'FAIL', &DSN, USERID1, USERID2, ...) -
          PNODE|SNODE)
```

The following example demonstrates a Process using the DMNOTFY2 program.

```

COPY   PROCESS  PNODE=CDA   SNODE=CDB
STEP01 COPY
      FROM      (PNODE
                DSN='FILE1 JCL'
                LINK=(VMID,PASS1,RR,125)
                DISP=SHR)
      TO        (SNODE
                DSN='FILE2 JCL'
                LINK=(VMID2,PASS2,MR,125)
                DISP=SHR )
STEP02 IF      (STEP01=0) THEN
NOTIFY RUN TASK
(PGM=DMNOTFY2,PARM=(CL4'GOOD','FILE1 JCL',CDID1,CDID2))SNODE
      ELSE
NOTIFY2 RUN TASK
(PGM=DMNOTFY2,PARM=(CL4'FAIL','FILE1 JCL',CDID1,CDID2))PNODE
EIF

```

A GOOD value would be relayed to the VM/ESA users CDID1 and CDID2 on CDB in the following message.

```

SNTA001I **CONNECT:Direct FILE TRANSFER SUCCESSFUL**
          TRANSFER BETWEEN - CDA          AND - CDB
          DSN=FILE1 JCL

```

A FAIL value would be relayed to the VM/ESA users CDID1 and CDID2 on CDA in the following message.

```

SNTA002I ***** CONNECT:Direct FILE TRANSFER FAILED *****
          ***** CONNECT:Direct FILE TRANSFER FAILED *****
          TRANSFER BETWEEN - CDA          AND - CDB
          DSN=FILE1 JCL
          ***** CONNECT:Direct FILE TRANSFER FAILED *****
          ***** CONNECT:Direct FILE TRANSFER FAILED *****

```

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 for VM/ESA load library.

Completion Code

When multiple actions are contained in the parameters, the completion code returned is the highest completion code encountered.

Dynamic Allocation Parameters

The RUN TASK command invokes DMRTDYN with parameters that define allocation actions to be executed.

Dynamic Allocation Guidelines

The following rules apply to the syntax of the parameters:

- ▶ You must specify the same DDNAME for each DMRTDYN pair.

WARNING: If you do not specify the same DDNAME for each DMRTDYN pair, you may obtain unpredictable results.

- ▶ These are the following DMRTDYN pairs:

- ALLOC/UNALLOC
- ALLOC/RELEASE
- ACCESS/RELEASE

Note: You must perform an UNALLOC or RELEASE function for every time you use the ALLOC or ACCESS function.

- ▶ Separate each unique allocation action by a parameter containing a fullword of minus one (F'-1').
- ▶ A leading blank on the first parameter is not required; however, there *must* be a leading blank on all subsequent parameters. If there is no blank, the program will issue an RC=30 and an SRTA005I message for the Process.
- ▶ Parameters must not extend past column 72, or they will not be read.

Description of the DMRTDYN Utility

Review the *Dynamic Allocation Guidelines* section on page 12-5 before performing any of the DMRTDYN functions described in the following table.

Job Parameter	Description
ACCESS	Accesses a specified minidisk. ACCESS is usually paired with the RELEASE function.
ALLOC	Allocates a file. The allocation request can contain most job stream DD statement options, including DCB options applicable to allocation. This is the default. ALLOC is usually paired with UNALLOC function.
UNALLOC	Unallocates a file. UNALLOC is usually paired with the ALLOC function.
RELEASE	Releases the minidisk where the file is located. RELEASE can be paired with either the ACCESS or ALLOC function.

Sample Program Uses

The following example demonstrate how you might use the functions of the DMRTDYN program.

Sample Processes Using DMRTDYN

The following examples illustrate how you can use the DMRTDYN program. In the first example, the DMRTDYN program is used in a ACCESS, COPY, and RELEASE Process.

```
CDACCESS  PROCESS SNODE=CSD.VM.CD09 NOTIFY=USER1
STEP1     RUN TASK (PGM=DMRTDYN,PARM=(
          C' ACCESS LINK=(BARGOR,ALL,RR,191) '
          C' DDNAME=INFILE ' ))
STEP2     COPY     FROM (DSN='PROFILE EXEC'
          LINK=(USER1,ALL,RR,191))
          TO (DSN='!SPOOL USER1 TEST DATA')
STEP3     RUN TASK (PGM=DMRTDYN,PARM=(
          C' RELEASE DDNAME=INFILE ' ))
```

Note: In the previous example, the same DDNAME must be used for both the ACCESS and RELEASE function.

In the next example, the DMRTDYN program is used in an ALLOC, COPY, and UNALLOC Process.


```

WORK1      PROCESS SNODE=CSD.VM.CD09 NOTIFY=USER1
STEP1      RUN TASK (PGM=DMRTDYN,PARM=(
           C' ALLOC DSN="PROFILE EXEC" '
           C' DDNAME=INFILE '
           C' LINK=(BARGOR,ALL,RR,191) '
           C' DISP=OLD '
           C' DCB=DSORG=PS ' ))
STEP2      COPY      FROM (DSN='PROFILE EXEC'
           LINK=(BARGOR,ALL,RR,191) '
           LINK=(USER1,ALL,RR,191))
           TO (DSN='!SPOOL USER1 TEST DATA')
STEP3      RUN TASK (PGM=DMRTDYN,PARM=(
           C' RELEASE DDNAME=INFILE ' ))

```

File Conversion Programs (DMRTPOMV and DMRTPOVM)

Two RUN TASK programs, DMRTPOMV and DMRTPOVM, allow the sending of partitioned data set (PDS) members to and from VM. Both programs accept a PDS with an 80-byte record length, and both programs run on MVS only.

MVS-to-VM File Conversion Using DMRTPOMV

DMRTPOMV converts an MVS partitioned data set to a sequential file to be sent to VM. The sequential file can then be sent to a MACLIB or TXTLIB in VM. LOADLIBS are not supported.

The only parameter to the DMRTPOMV program is the input MVS PDS filename, followed by a blank, followed by the output sequential file name. You must predefine the output file. Enclose the entire parameter in single quotes.

The following figure shows an example of a file conversion from MVS to VM using DMRTPOMV.

```

RUN TASK (PGM=DMRTPOMV
           PARM=('MVS.PDS.NAME OUTPUT.SEQUENTIAL.FILE'))
COPY      FROM
           (PNODE DSN=OUTPUT.SEQUENTIAL.FILE
           DISP = SHR)
           TO
           (SNODE DSN='VM MACLIB'
           LINK = (SMITH,BOB,MR,125)
           DISP = RPL)

```

VM-to-MVS File Conversion Using DMRTPOVM

DMRTPOVM converts a VM MACLIB or TXTLIB that has been received from a VM node to an MVS partitioned data set.

The only parameter is the input CMS sequential file name, followed by a blank, followed by the output MVS PDS file name. You must predefine the MVS PDS output file. Enclose the entire parameter in single quotes.

The following figure shows an example of a file conversion from MVS to VM using DM RTPOMV.

```
COPY FROM
(SNODE DSN='VM MACLIB'
LINK=(SMITH,BOB,RR,125)
DISP=SHR)
TO
(PNODE DSN=CMS.SEQUENTIAL.FILE
DISP=RPL)
RUN TASK (PGM=DMRTPOMV
PARM=('CMS.SEQUENTIAL.FILE MVS.PDS.NAME'))
```

Punch a File Program (DMGPUN01)

The RUN TASK program, DMGPUN01, enables you to PUNCH a file, containing one record to the machine of your choosing. In addition, you may also specify the CLASS of the file and provide 64 bytes of your own unique information to the file.

To execute this command, type in the following syntax:

```
RUN TASK (PGM=DMGPUN01 PARM=('YOUR TEXT',USERID,CLASS))-
PNODE|SNODE
```

If your submitted text consists of more than one word, then it must be enclosed in either single (') or double (") quotes. When using double quotes, symbolic variables within the text will be substituted.

Understanding the DMGPUN01 Program

In this example, the file is sent to SERVER01 as a Class C file.

```
NOTE1 RUN TASK (PGM=DMGPUN01
PARM=('COMP001 Production database transfer',
SERVER01,C)) PNODE
```

In the previous example, the filename will be C:DPROC and the filetype will be composed of an S or P for the first character, indicating the RUN

TASK was executed on the PNODE or the SNODE, followed by the Process number. The previous example appears as:

```
COMP0001 Production database transfer ../.. P046335G00000000
```

When SNODE is specified, the highest completion code field will always be zero because the Process information is sent one step at a time from the PNODE to the SNODE as a Process is executed. Therefore, the highest return code on the SNODE is the DMGPUN01 RUN TASK step.

Note: Only the PNODE has a complete record of previous steps and their return codes.

If you code multiple RUN TASK steps for the DMGPUN01 program, the resulting files that are punched will all have the same filename and filetype. The filename is always C:DPROC and the filetype is either S (for SNODE) or P (for PNODE) concatenated with the Process number which does not change between the steps within a Process.

The following table outlines information available to the receiver of the file:

Available Information	Location
Origin userid of file	Obtained from the RDR file
Run on PNODE or SNODE	Obtained from the RDR filetype
Process number	Obtained from the RDR filetype
Date of execution	Located from the date on the RDR file
Time of execution	Located on the time on the RDR file
User's unique text	Obtained from the file contents
Process name	Obtained from the file contents
Highest completion code	Obtained from the file contents

The following EXEC illustrates the receiving and parsing of the data contained within the DMGPUN01 file.

```
/* CDREC EXEC 12/15/97 */
Trace 'e'
RDR. =''
/* Using FULLDATE option for 4-digit year requires VM 2.2.00 */
'PIPE CP QUERY RDR ALL FULLDATE |', /* Class G user QUERY */
'LOCATE 59-66 /C:DPROC/ |'. /* Locate only this filename */
'STEM RDR.'
say 'Files in rdr =' rdr.0
Do R = 1 RDR.0
If RDR.r ='' then leave R; /* 2nd check. RDR.0 should limit */
*/
Parse var RDR.r 1 rdr_origin 9 . 10 rdr_number 14 . 15 rdr_class,
16 . 39 rdr_date 49 . 50 rdr_time 59 . 69 rdr_type 77.
Parse var rdr_type 1 origin 2 procnum
PIPE 'CMS RECEIVE' rdr_number '= = K ( REP | STEM CRDR.'
Parse var CDRDR.2 . fn ft fm other
'EXECIO 1 DISKR' fn ft fm '(FINIS VAR NOTIFY'
Parse var NOTIFY 1 housemsg 65 PROCNAME 73 HICOMP

say' Message from:' RDR_ORIGIN
say' My message data:' housemsg
say' The process name:' PROCNAME
say' The process number:' PROCNUM
say' RUN on node (P/S):' ORIGIN
say' Highest completion code:' HICOMP
say' RUN TASK run date:' RDR_DATE
say' RUN TASK run time:' RDR_TIME
say;
/* Take specific action here */
/* Take specific action here */
/* Take specific action here */
'ERASE' fn ft fm /* Done. Get rid of the file */
End;
Exit;
```

Glossary

A

ADJACENT.NODE

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 (Application Program Interface)

This CONNECT:Direct component 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 applid.

Authorization Facility

CONNECT:Direct facility that grants access to CONNECT:Direct commands.

Authorization File

CONNECT:Direct for VM/ESA authorization file used to control access to CONNECT:Direct for VM/ESA and identify commands that can be executed by userid. This file can also be used in conjunction with security exit interfaces with the secured point of entry feature.

B

Batch Interface

CONNECT:Direct interface that allows users to request CONNECT:Direct services from a batch job stream through control statements passed to a CONNECT:Direct-supplied program, DMBATCH.

C

Change Process Command

CONNECT:Direct command that changes CONNECT:Direct Process parameters of Processes in the TCQ.

Checkpoint/Restart

Eliminates the need to retransmit an entire file in the event of a transmission failure. A value on the COPY statement or in the initialization parameter, CKPT.MODE and CKPT, specifies the checkpoint interval. If a copy procedure is interrupted, CONNECT:Direct will restart that copy at the last checkpoint.

Command Line Interface

CONNECT:Direct interface that allows users to submit CONNECT:Direct Processes and commands from their native command line environment.

Commands

Are used to initiate and monitor activity within the CONNECT:Direct system and can be issued from the ISPF IUI, the operator console, a batch job, or a user application program.

CONNECT:Direct

Family of data transfer software products that distribute information and manage production activities among multiple data centers. CONNECT:Direct for VM/ESA runs as a VTAM application in a VTAM/GCS environment.

CONNECT:Direct Commands

Are used to initiate and monitor activity within the CONNECT:Direct system and can be issued from the ISPF IUI, the operator console, a batch job or a user application program.

CONNECT:Direct Process

A series of statements (which can be predefined and stored in a library) submitted through the API to initiate activity, such as copying files, running jobs, and so on.

Connection Type

Installation parameter that determines the session type, in order of preference, that a CONNECT:Direct node will use to establish communications between LUs (logical units). Valid values are LU0, LU2, and LU62.

COPY Statement

CONNECT:Direct Process statement used to copy a file from one node to another.

Cross Domain

Pertaining to control of resources involving more than one domain. A domain consists of one SSCP (System Services Control Point) and all the SNA resources under its control. This has meaning when performing a multiple session signon to another CONNECT:Direct DTF, as well as establishing a connection to another CONNECT:Direct DTF for purposes of copying files, etc.

D

DATEFORM initialization parameter

The DATEFORM initialization parameter specifies the order of the Gregorian numeric values for month (M), day of the month (D), and year (Y). The default value is MDY.

Delete Process Command

CONNECT:Direct command that deletes a non-executing CONNECT:Direct Process from the TCQ.

DTF (Data Transmission Facility)

The nucleus component of CONNECT:Direct. The DTF controls information distribution to other CONNECT:Direct nodes in the network. Start-up parameters that govern the overall activity of the DTF are defined within the initialization parameters.

E

ELSE Statement

CONNECT:Direct modal statement that designates a block of CONNECT:Direct statements that will execute when the IF THEN condition is not satisfied.

EIF Statement

CONNECT:Direct modal statement that specifies the end of an IF THEN or IF THEN ELSE block of CONNECT:Direct Process statements.

EXIT Statement

CONNECT:Direct modal statement that can be used to bypass all remaining steps within a CONNECT:Direct Process.

F

Flush Process Command

CONNECT:Direct command that stops an executing CONNECT:Direct Process.

G

GOTO Statement

CONNECT:Direct modal statement that allows bypassing of steps within a CONNECT:Direct Process.

Gregorian dates

Gregorian date format consists of numerical values for the month, day of the month, and the year. You must use the DATEFORM initialization parameter to specify the Gregorian values that CONNECT:Direct uses to process Gregorian dates.

I

IF THEN Statement

CONNECT:Direct modal statement that allows executing of a block of CONNECT:Direct statements based on the completion code of a previous Process step.

IUI (Interactive User Interface)

The IUI interface is 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.

J

Julian dates

Julian date format consists of numeric values for the year and the day of the year. The Julian day of the year value always follows the Julian year value. The Julian day of the year value is always a three-digit numeric with a range from 1–366. For example, 017 is the Julian day of the year value for January 17.

L

LOCAL.NODE

Entry in the Network Map. This entry represents the CONNECT:Direct logical node name and its associated VTAM applid for the local copy of CONNECT:Direct for VM/ESA. There is only one LOCAL.NODE entry in each Netmap.

M

Max PNODE Sessions

The maximum number of concurrent PNODE (local process execution) sessions that this CONNECT:Direct node is to allow. This is a installation parameter used to build the Network map.

MAX.PRIMARY

Specifies maximum number of primary node-to-node sessions that will be allowed to start on a node. A CONNECT:Direct primary node represents the node-to-node half-session that was started as the result of one or more Processes being ready to run. The CONNECT:Direct primary node is the session initiator or PNODE. The range for MAXPRIMARY is from 2-512.

MAX.RETRIES

The maximum number of times that this CONNECT:Direct node is to retry execution of a Process that is suspended due to a recoverable error. This is a initialization parameter. Also see Retry Interval.

MAX.SESIONS

Defines the maximum number of concurrent sessions (DTF-to-DTF) that this particular CONNECT:Direct node is to allow. This is a installation parameter used to build the Network Map.

MAX.SECONDARY

Specifies the maximum number of secondary node-to-node sessions that will be allowed to start on a node. A CONNECT:Direct secondary node represents the node-to-node half-session that was started because of primary node initiation. The range for MAXSECONDARY is from 2-512.

MAX.TAPE

Specifies the maximum number of tape processes that will be allowed to start in a node. The range of acceptable values for MAX.TAPE is 1-32767.

Modal Statements

CONNECT:Direct modal statements (IF THEN, EIF, ELSE, EXIT, and GOTO) allow you to alter the sequence of CONNECT:Direct Process execution based on completion of a previous Process step.

Modify Command

CONNECT:Direct command that allows you to request traces or modify certain CONNECT:Direct operational functions.

Multiple Session Signon

Command issued through the API that allows the API user to sign on to a remote DTF.

N

Network Map

VSAM file identifying all valid CONNECT:Direct nodes and applids in the network. There is one Network Map (Netmap) associated with each CONNECT:Direct for VM/ESA node. There is one entry within that netmap for each of the other CONNECT:Direct nodes to which the local CONNECT:Direct for VM/ESA node can initiate a session. The netmap entries also contain the rules or protocol to which the nodes will adhere when communicating.

Node

Any site within a network from which information distribution can be initiated.

O

Online Messages

Completion and error messages that are displayed online.

Operator Interface

Allows CONNECT:Direct commands to be issued from the VM/ESA operator console. This interface also allows tailoring of CONNECT:Direct commands through a command list (CLIST) facility.

P

Parallel Sessions

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.

PNODE (Primary Node)

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

Process statements 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).

R

Retry Interval

Installation parameter that specifies the interval, in minutes, that the retries mentioned in the Max Retries parameter will be performed.

RUN JOB Statement

Executed as a CONNECT:Direct Process statement, allowing a job to be submitted to the VM/ESA internal reader.

RUN TASK Statement

Allows an external program to be executed under the control of CONNECT:Direct.

S

Select Netmap Command

CONNECT:Direct command that allows you to reference the definitions of nodes that you can communicate with.

Select Process Command

CONNECT:Direct command that allows you to view CONNECT:Direct Processes that are in the Transmission Control Queue (TCQ).

Select Statistics Command

CONNECT:Direct command that displays or prints statistics from the statistics log.

Session Classes

Installation parameter that specifies the process class groupings, priorities and number of Processes that can be concurrently executed on this CONNECT:Direct node.

Signoff Command

CONNECT:Direct command that terminates a CONNECT:Direct API connection with the DTF.

Signon Command

CONNECT:Direct command that initiates a CONNECT:Direct API connection with the DTF.

SNODE (Secondary Node)

The CONNECT:Direct node that interacts with the primary node (PNODE) during process execution. The secondary node (SNODE) may also be referred to as the participating, target or destination node. In every Process, there is one PNODE and one SNODE.

Statistics File

VSAM relative record data set that holds CONNECT:Direct statistics records to document the history of a process. This is a wrap-around file.

Statistics Facility

CONNECT:Direct facility that records all CONNECT:Direct activities.

STOP NDM Command

CONNECT:Direct command that stops CONNECT:Direct-DTF operation.

Submit Command

CONNECT:Direct command that submits a CONNECT:Direct Process for execution.

SUBMIT Statement

CONNECT:Direct command that causes another CONNECT:Direct Process to be submitted when this statement is encountered in an executing Process.

SUPERUSR Password

Password reserved for the SUPERUSR userid. The password can be from 1-8 characters. Also see SUPERUSR ID.

SUPERUSR ID

Special ID that is not required to pass through normal security checking when signing on to CONNECT:Direct. This ID will allow you to perform CONNECT:Direct administrative functions.

Suspend Process Command

CONNECT:Direct command that suspends an executing CONNECT:Direct Process.

Swap Node Command

CONNECT:Direct command that allows you to swap to another CONNECT:Direct node when you are signed on to more than one CONNECT:Direct node. Also see Multiple Session Signon.

SYMBOL Statement

CONNECT:Direct Process statement that allows you to build symbolic substitution values.

Symbolics

Symbolic parameters are supported within CONNECT:Direct Processes. This allows 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.

T

TCP/IP

TCP/IP, Transmission Control Protocol/Internet Protocol, is a set of network standards that specify the details of how computers communicate, as well as a set of conventions for interconnecting networks and routing traffic.

TCQ (Transmission Control Queue)

A VSAM relative record data set (RRDS) used to hold all Processes that have been submitted to CONNECT:Direct.

Y

Year 2000 Compliance

Software requirements that ensure turn-of-the-century date processing, product functionality, security, and error recovery.

A

- accounting data
 - Pnode, 6-3
 - Snode, 6-4
- active session, 6-8
- ADMIN option, Primary Menu Option, 4-5
- API. *See* Application Program Interface
- Application Program Interface, 1-2
 - description, 1-4
 - used with user programs, 1-4
- ARCHDSN parameter, description, with SELECT STATISTICS command, 9-12
- asterisk, in command, 2-5
- AUTH option, User Authorization screen, 4-6

B

- backslashes, in commands, 2-9
- batch commands, requirements, 3-1
- Batch Interface, description, 1-3
- batch interface, definition, 2-1
- Batch Interface program
 - command requirements, 3-1
 - ddnames, 3-3
 - DMBATCH, 3-1
 - multiple session SIGNON, 6-7, 6-9
 - return codes, 3-4
 - SWAP NODE command, 6-11
 - with SIGNOFF command, 6-13
 - with SIGNON, 6-4
 - with SWAP NODE, 6-10

- batch processing
 - asynchronous, 3-2
 - sample EXEC, 3-3
 - SUBMIT command, 3-1
 - with DELETE PROCESS command, 8-12
 - with FLUSH PROCESS command, 8-12
 - with SELECT PROCESS command, 8-18
 - with SUSPEND PROCESS command, 8-12
- bold letters, notational conventions, xvi
- bracketing for special characters, in commands, 2-9
- brackets, notational conventions, xvi
- branching
 - from command line, 4-7
 - from summary displays, 4-9
 - to Administrative screens, 4-8
 - to ISPF, 4-8
- browsing table displays, 2-12, 4-9

C

- CASE parameter
 - description
 - with CHANGE PROCESS command, 8-4
 - with DELETE PROCESS command, 8-12
 - with FLUSH PROCESS command, 8-12
 - with SELECT PROCESS command, 8-18
 - with SELECT STATISTICS command, 9-11
 - with SUBMIT command, 7-3
 - with SUSPEND PROCESS command, 8-12
 - general use, 2-9
 - with SIGNON command, 6-2

- case sensitivity
 - at the command level
 - CHANGE PROCESS command, 8-4
 - DELETE PROCESS command, 8-12
 - FLUSH PROCESS command, 8-12
 - SUBMIT command, 7-3
 - SUSPEND PROCESS command, 8-12
 - at the session level, SIGNON command, 6-2
 - how to indicate, 2-9
- CCODE parameter, description, with SELECT STATISTICS command, 9-5
- CF option, Copy File, 7-17
- CHANGE PROCESS command
 - batch interface usage, 8-7
 - examples, 8-7
 - format, 8-3
 - IUI usage, 8-8
 - Change Process screen, 8-8
 - parameters
 - optional, 8-4
 - required, 8-3
 - purpose, 8-2
 - release from Hold Queue, 8-2
- Change Process screen, 8-8
- checkpoint position, restart, 8-6
- CLASS parameter, description, with SUBMIT command, 7-3
- CLI, definition, 2-2
- CMD prompt, 4-7
- CMS Command Line Interface, 5-1–5-2
 - definition, 2-2
 - example, 5-2
 - files defined, 5-1
 - Select Message command, 5-2
- Command Line Interface
 - CMS, 5-2
 - Select Message command, 5-2
- command line interface, description, 1-3
- command stacking, how to, 6-7
- commands
 - CHANGE PROCESS, 8-3
 - definition of, 2-1
 - DELETE PROCESS, 8-11
 - description of, 2-3
 - FLUSH PROCESS, 8-11
 - function, 1-5, 1-7
 - methods of issuing, 2-1
 - overview of, 2-2
 - SELECT NETMAP, 10-1
 - SELECT PROCESS, 8-15
 - SELECT STATISTICS, 9-4
 - SIGNOFF, 6-13
 - SIGNON, 6-2
 - SUBMIT, 7-2
 - supported, 2-2
 - SUSPEND PROCESS, 8-11
 - SWAP NODE, 6-10
 - symbolic substitution, 2-6
 - syntax, 2-5
 - SYSIN batch location, 3-3
 - writing, 2-3
- commas, in commands, 2-5
- comments
 - in commands, 2-7
 - syntax, 2-7
- completion, with SELECT STATISTICS command, CCODE parameter, 9-5
- compression percentage
 - calculation formula, 9-2
 - with statistics record, 9-2
- concatenation, in commands, 2-8
- CONNECT:Direct, description, 1-1
- Continuation marks
 - in commands, 2-6
 - with concatenation, 2-8
- Conversion Programs
 - with DMRTPOMV, 12-7
 - with DMRTPOVM, 12-7
- Copy File Menu, 7-19
- Copy Termination statistics record, content, 9-1
- Copyfile – Receiving File screen, VM-DISK, 7-22
- Copyfile – Sending File screen
 - MVS, 7-21, 7-22
 - OS/400, 7-21
- copyright screen, 6-5
- CP option, through the IUI, 8-8

D

- Data Transmission Facility (DTF), description. *See* DTF
- dates
 - how CONNECT:Direct processes, 7-8, 9-6, 9-8
 - specifying, 7-8, 9-6, 9-8

- defaults at signon, 6-7
- define a Process, 7-14
- delete a nonexecuting Process, from Selected Process screen, 8-23
- Delete a Nonexecuting Process screen, 8-15
- DELETE PROCESS command
 - batch interface usage, 8-12
 - examples, 8-13
 - format, 8-11
 - IUI usage, 8-14
 - Delete a Nonexecuting Process screen, 8-15
 - parameters
 - optional, 8-12
 - required, 8-11
 - purpose, 8-9
- delete session, 6-12
- delimiters, in commands, 2-8
- DEST parameter, description
 - with CHANGE PROCESS command, 8-4
 - with SELECT PROCESS command, 8-16
- DF option, Define Process, 7-14
- DISPLAY parameter
 - general use, 2-11
 - in IUI, 4-9
- DISplay parameter, description, with SELECT NETMAP command, 10-2
- displaying output on screen, 2-11
- displaying your signon defaults, 6-6
- DMBATCH
 - batch command submission, 3-1
 - ddnames, 3-3
 - DELETE PROCESS command, 8-12
 - FLUSH PROCESS command, 8-12
 - parameter descriptions, 3-4
 - processing rules, 3-2
 - requirements, 3-1
 - return codes, 3-4
 - sample EXEC, 3-3
 - SELECT NETMAP command, 10-2
 - SELECT PROCESS command, 8-18
 - SIGNOFF command, 6-11, 6-13
 - SUBMIT command, 7-11, 8-7
 - SUSPEND PROCESS command, 8-13
 - syntax, 3-1, 3-2
- DMCHLAPI, with CMS Command Line Interface, 5-1
- DMCHLAPI, with DMBATCH, 3-4

- DMMSGFL, ddname in batch job, 3-3
- DMNETMAP, ddname in batch job, 6-3
- DMNOTIFY, operator console notify program, 12-2
- DMPRINT, ddname in batch job, 3-3
- DMRTDYN
 - completion code, 12-5
 - dynamic allocation program, 12-4
 - parameters, 12-5
 - sample program uses, 12-6
- DMRTPOMV
 - file conversion example, 12-7
 - File Conversion Program, 12-7
 - parameters, 12-7
- DMRTPOVM
 - file conversion example, 12-7
 - File Conversion Program, 12-7
 - parameters, 12-8
- DMNOTFY2, user notify program, 12-3
- DP option, Delete Process screen, 8-14
- DSN, parameter with SUBMIT command, 7-3
- DTF. *See* Data Transmission Facility
- dynamic allocation program, DMRTDYN, 12-4

E

- elapsed time
 - significance, 9-3
 - with statistics record, 9-3
- End, use as IUI SIGNOFF, 6-14
- EXCLUDE parameter, description, with SELECT STATISTICS command, 9-5
- EXEC, startup, 6-3

F

- File Conversion Programs
 - with DMRTPOMV, 12-7
 - with DMRTPOVM, 12-7
- FILE parameter
 - description
 - with SELECT PROCESS command, 8-18
 - with SELECT STATISTICS command, 9-12
 - general use, 2-11

files, defined with CMS Command Line Interface, 5-1

filetype, NDMPROC, 7-3

FLUSH PROCESS, for SNODE Run Task, 8-10

FLUSH PROCESS command

batch interface usage, 8-12

examples, 8-13

format, 8-11

IUI usage, 8-14

Flush Process screen, 8-15

parameters

optional, 8-12

required, 8-11

purpose, 8-9

Flush Process screen, 8-15

flush the currently executing Process, from Selected Process screen, 8-23

FNAME parameter, description, with SELECT STATISTICS command, 9-11

FOLD, parameter with SIGNON, 6-2

folding to upper case, in printouts, 6-2

FORCE FLUSH, with MP=YES, 8-10

FORCE parameter, description

with FLUSH PROCESS command, 8-12

with SUSPEND PROCESS command, 8-12

FORCE SUSPEND, with MP=YES, 8-10

FP option, Flush Process screen, 8-14

function keys, 4-8

G

GCSAPI, parameter with SIGNON, 6-2

generic specification, use of asterisk, 2-5

GLOBAL LOADLIB command, with CMS Command Line Interface, 5-2

H

HELP/PF1, online help, 4-7

HOLD parameter, description

with CHANGE PROCESS command, 8-4

with SUBMIT command, 7-3

Hold Queue

with CHANGE PROCESS command, 8-2, 8-7

with SUSPEND PROCESS command, 8-9

hyphen, as a continuation mark, 2-6

I

ID

security exit on PNODE, 7-5

security exit on SNODE, 7-7

security with SIGNON, 6-4

indicating

case sensitivity, 2-9

output destination, 2-11

selection criteria, 2-10

initial command, SIGNON default, 6-7

Interactive User Interface, description, 1-2

interactive user interface, definition, 2-2

Internal Components, Data Transmission Facility, 1-2

internal components

Application Program Interface, 1-2

interfaces, 1-2

internal components of CONNECT:Direct, 1-1

IUI

definition, 2-2

multiple session SIGNON, 6-9

Operator Table, 4-10

SELECT Process, 8-18

SIGNOFF, 6-14

SIGNON, 6-5

SUBMIT, 7-12

SWAP NODE, 6-11

IUI Screen Commands

branching from the command line, 4-7

messages, 4-11

online help, 4-7

PF Keys, 4-8

J

job

batch destination of output, 3-3

submit through batch, 7-11

K

keyword, definition, 2-4

L

label, in native format, 2-4
license violation, detection, 6-3
list delimiter, 2-5
lowercase letters, notational conventions, xvi

M

M command, 4-11
MACLIB, Process location, 7-3
master session
 description, 6-7
 termination, 6-15
MAXDELAY, for synchronizing submitted processes, 3-5
MAXDELAY parameter, description, with SUBMIT command, 7-4
Member Copy statistics record, content, 9-1
Message Display screen, 4-12, 4-13
message maintenance options, 4-12
Message Maintenance screen, 4-12
 option 1, 4-13
 option 2, 4-13
 options 3 and 4, 4-14
Message Print screen, 4-14
Message Summary screen, 4-13
messages
 CONNECT:Direct generated, 4-11
 file location in batch job, 3-3
 ISPF/PDF generated, 4-11
 viewing in IUI, 4-11
MODAL statements, with User Notification Programs, 12-1
monitor the currently executing Process, from Selected Process screen, 8-23
MS option, Multiple Session Signon screen, 6-9
MSG option, Message Maintenance screen, 4-12
Multi-session signon, non-master session, parameter list, 6-8
Multi-session signon
 master session, 6-7
 parameter list, 6-7
Multiple Session Signon, screen, 6-10

Multiple Session SIGNON Command
 batch interface usage, 6-9
 IUI usage, 6-9
 master session, 6-7
 switching nodes, 6-8

N

native format
 definition of, 2-4
 structure of, 2-4
 use of commands, 2-4
 use of label, 2-4
 use of parameters, 2-4
 use of subparameters, 2-4
NDMPROC, filetype with SUBMIT command, 7-3
NETMAP, parameter with SIGNON, 6-3
NEWNAME parameter, description, with SUBMIT command, 7-4
NM option, Select Network Map screen, 10-3
No Process Found message, Operator Table, 8-21
NODE, parameter with SIGNON, 6-3
node, remote SIGNON, 6-4
NODE parameter, description, with Select Netmap command, 10-2
nodes
 swapping, 6-10
 switching, 6-8
notational conventions
 bold letters, xvi
 brackets, xvi
 lowercase letters, xvi
 miscellaneous, xvii
 underlined letters, xvi
 uppercase and lowercase letters, xvi
 uppercase letters, xvi
 vertical bars, xvi
NOTIFY parameter, description, with SUBMIT command, 7-5

O

online help, 4-7
operator console notify program, DMNOTIFY, 12-2
operator Interface, description, 1-3

Operator Table – Select Process
 accessing, 8-20
 accessing Execution Queue Status using, 8-21
Operator Table / Executing Queue, 8-22
Operator Table screen, 8-20
operator tables
 general characteristics, 4-10
 issuing commands from, 4-10
output, batch destination, 3-3
output type selection, 2-11
Overview, Of this manual, xiv

P

PACCT, parameter with SIGNON, 6-3
PACCT parameter, with SUBMIT command, 7-5
parameters
 DMRTDYN, 12-5
 in DMBATCH EXEC statement, 3-4
 in native format, 2-4
 with DMRTPOMV, 12-7
 with DMRTPOVM, 12-8
parentheses, in commands, 2-6
password
 new with SIGNON, 6-4
 security with SIGNON, 6-4
PF Keys, 4-8
PF1 key, online help, 4-7
PNAME parameter, description
 with CHANGE PROCESS command, 8-4
 with DELETE PROCESS command, 8-11
 with FLUSH PROCESS command, 8-11
 with SELECT PROCESS command, 8-16
 with SELECT STATISTICS command, 9-6
 with SUSPEND PROCESS command, 8-11
Pnode, accounting data with SIGNON, 6-3
PNODE parameter, description, with SUBMIT
 command, 7-5
PNODEID parameter, description, with SUBMIT
 command, 7-5
PNUMBER parameter, description
 with CHANGE PROCESS command, 8-4
 with DELETE PROCESS command, 8-11
 with FLUSH PROCESS command, 8-11
 with SELECT PROCESS command, 8-16
 with SELECT STATISTICS command, 9-6
 with SUSPEND PROCESS command, 8-11
positional parameters, definition, 2-4
Primary Options Menu
 bypassing, 6-7
Options
 AUTH, 4-6
 CF, 7-17
 CP, 8-8
 DF, 7-14
 DP, 8-14
 FP, 8-14
 MS, 6-10
 MSG, 4-12
 NM for NETMAP command, 10-3
 PS, 8-14
 SB, 7-13
 SD, 6-6
 SIGNOFF, 6-14
 SP, 8-19
 SS, 9-16
 SUS, 8-14
 SW, 6-11
 WHO, 6-6
 screen sample, 4-1
PRINT parameter
 general use, 2-11
 when it is used, 2-11
Print parameter, description
 with SELECT NETMAP command, 10-2
 with SELECT PROCESS command, 8-18
 with SELECT STATISTICS command, 9-12
printing, Message text, 4-12
printing output, 2-11
printouts, case selection, 6-2
PROC, parameter with SUBMIT command, 7-3
Process
 control, 8-22
 control through commands, 8-1
 defaults with SIGNON, 6-3
 definition of, 1-5
 DELETE PROCESS command, 8-9
 example
 monitoring, 1-8
 multi-step, 1-7
 review statistics, 1-8
 submitting to TCQ, 1-7
 FLUSH PROCESS command, 8-9
 SUSPEND PROCESS command, 8-9
Process control commands, purpose, 8-1

- Process Definition (DF), 7-14
- Process Definition screen, 7-14
- Process release from hold queue, from Selected Process screen, 8-23
- Process statements, in batch jobs, 3-1
- Process Submit statistics record, content, 9-1
- Process Termination statistics record, content, 9-1
- Processes, determining the outcome, 1-8
- PROFILE GCS, startup exec, 6-3
- PRTY parameter, description
 - with CHANGE PROCESS command, 8-5
 - with SUBMIT command, 7-6
- PS option, Suspend Process screen, 8-14
- put a nonexecuting Process in the Hold Queue, from Selected Process screen, 8-23

Q

- queue, TCQ (Transmission Control Queue), 1-6
- QUEUE parameter, description, with SELECT PROCESS command, 8-16
- quotation marks, in commands, 2-7

R

- re-establish session, SIGNON command, 6-3
- RECONNECT, parameter with SIGNON, 6-3
- release a Process from the Hold Queue, from Selected Process screen, 8-23
- RELEASE parameter, description, with CHANGE PROCESS command, 8-5
- REQUEUE parameter, description, with SUBMIT command, 7-6
- RESTART parameter, description, with CHANGE PROCESS command, 8-5
- RETAIN parameter, description
 - with CHANGE PROCESS command, 8-6
 - with SUBMIT command, 7-6
- Run Job statistics record, content, 9-1
- RUN TASK programs
 - with DMRTPOMV, 12-7
 - with DMRTPOVM, 12-7
- RUN TASK record fields, PRIMARY, 9-2

- RUN TASK statements, with user notification programs, 12-1
- Run Task statistics record, content, 9-1

S

- S option, on summary displays, 4-9
- SACCT, parameter with SIGNON, 6-4
- SACCT parameter, description, with SUBMIT command, 7-7
- saving output, 2-11
- SB option, through the IUI, 7-13
- screens
 - Change Process, 8-8
 - Copy File Menu, 7-19
 - Copyfile – Receiving File (VM-DISK), 7-22
 - Copyfile – Sending File (MVS), 7-21, 7-22
 - Copyfile – Sending File (OS400), 7-21
 - Message Display, 4-12, 4-13
 - Message Maintenance, 4-12
 - Message Summary, 4-13
 - Multiple Session Signon, 6-10
 - Operator Table, 8-20
 - Operator Table / Executing Queue, 8-22
 - Primary Options Menu, 4-1
 - Process Definition, 7-14
 - Security Override
 - Copy File, 7-20
 - Submit Process, 7-14
 - Select Network Map or TCP Information, 10-3
 - Select Process, 8-18, 8-19
 - Select Process Display Output, 8-19
 - Select Statistics, 9-16
 - Selected Process, 8-22
 - Signon Defaults, 6-6
 - Statistics Detail – Copy, 9-18
 - Statistics Summary, 9-17
 - Submit Process, 7-13
 - Suspend an Executing Process, 8-14
 - Swap/Display User Sessions, 6-11
 - User Authorization, 4-6
 - User Information (WHO), 6-6
- SD option, through the IUI, 6-6
- security, USERID parameter, 6-4
- security at signon, 6-1
- Security Override screen
 - Copy File, 7-20
 - Submit Process, 7-14

- Select Message command, Command Line Interface, 5-2
- SELECT NETMAP command
 - batch interface usage, 10-2
 - format, 10-1
 - IUI usage, 10-3
 - parameters
 - optional, 10-2
 - required, 10-2
 - purpose, 10-1
- Select Netmap command, example, 10-3
- Select Network Map or TCP Information screen, 10-3
- SELECT PROCESS command
 - accessing Operator Table through IUI, 8-20
 - batch interface usage, 8-18
 - format, 8-15
 - IUI usage, 8-18
 - Operator Table, 8-20
 - output format, 8-19
 - Select Process screen, 8-18, 8-19
 - parameters
 - optional, 8-18
 - required, 8-16
 - purpose, 8-15
- Select Process Display Output screen, 8-19
- Select Process screen, 8-18, 8-19
- SELECT STATISTICS command
 - batch interface usage, 9-12
 - display unformatted records, 9-16
 - examples, 9-13
 - format, 9-4
 - IUI usage, 9-16
 - Statistics Display, 9-18
 - Statistics Summary, 9-17
 - parameters
 - optional, 9-12
 - required, 9-4
 - purpose, 9-3
 - record types, 9-1
 - sample output, 9-15
- Select Statistics screen, 9-16
- Selected Process options, 8-23
- Selected Process screen, 8-22
- selection criteria, where parameter, 2-10
- session
 - maximum number, 6-9
 - re-establish VTAM, 6-3
- Session Case Sensitive field, SIGNON screen, 6-2
- sessions
 - currently active, 6-12
 - delete, 6-12
 - signoff sequence, 6-8
 - swap, 6-12
- SIGNOFF command
 - batch interface usage, 6-13
 - format, 6-13
 - IUI usage, 6-14
 - purpose, 6-13
- SIGNON
 - batch interface example, 6-4
 - batch interface usage, 6-4
 - from remote node, 6-4
 - viewing your current parameters, 6-6
 - with CMS Command Line Interface, 5-1
- signon
 - different nodes, 6-14
 - to CONNECT:Direct for MVS, 6-5
- SIGNON command
 - format, 6-2
 - parameter descriptions, 6-2
 - purpose of, 6-1
- signon control, 6-1
- signon defaults
 - display how currently set, 6-6
 - options, 6-7
 - setting up, 6-6
- Signon Defaults screen, 6-6
- Snode, accounting data with SIGNON, 6-4
- SNODE parameter, description, with SELECT STATISTICS command, 9-10
- SNODE parameter, description, with SUBMIT command, 7-7
- SNODEID parameter, description, with SUBMIT command, 7-7
- SOPA006I message, 8-21
- SP option, Select Process screen, 8-18
- special characters, in commands, 2-8
- specifying dates, into Year 2000, 7-8, 9-6, 9-8
- SPF option, Primary Menu Option, 4-8
- SS option, Select Statistics screen, 9-16
- STARTT parameter, description
 - with CHANGE PROCESS command, 8-7

- with SELECT STATISTICS command, 9-6
- with SUBMIT command, 7-8
- startup exec, PROFILE GCS, 6-3
- Statistics Detail – Copy screen, 9-18
- Statistics Display screen, from Select Statistics screen, 9-18
- Statistics Log, record types, 9-1
- Statistics Summary screen, 9-17
 - from Select Statistics screen, 9-17
- STATUS parameter, description, with SELECT PROCESS command, 8-17
- STOPT parameter, description, with SELECT STATISTICS command, 9-8
- submit, synchronizing with MAXDELAY, 3-5
- SUBMIT command
 - batch interface usage, 7-11
 - examples, batch, 7-11
 - format, 7-2
 - IUI usage, 7-12
 - parameters
 - optional, 7-3
 - required, 7-2
 - purpose, 7-2
 - submitting a Process, through the IUI, 7-12
- Submit Process screen, 7-13
- Submit statement within a Process statistics record, content, 9-1
- SUBmitter parameter, description
 - with CHANGE PROCESS command, 8-4
 - with DELETE PROCESS command, 8-12
 - with FLUSH PROCESS command, 8-12
 - with SELECT PROCESS command, 8-17
 - with SUSPEND PROCESS command, 8-12
- submitting a Process, 7-21, 7-22
 - create COPY Process, 7-17
 - predefined, 7-13
 - through the IUI, 7-12
 - using the SUBMIT command, 7-2
- subparameters
 - definition, 2-5
 - in native format, 2-4
- SUSPEND PROCESS, for SNODE Run Task, 8-10
- SUSPEND PROCESS command
 - batch interface usage, 8-12
 - examples, 8-13
 - format, 8-11

- IUI usage, 8-14
 - Suspend Process screen, 8-15
- parameters
 - optional, 8-12
 - required, 8-11
- purpose, 8-9
- Suspend Process screen, 8-15
- suspend the currently executing Process, from Selected Process screen, 8-23
- SW option, Swap / Display User Sessions screen, 6-11
- SWAP NODE command
 - batch interface usage, 6-11
 - format, 6-10
 - IUI usage, 6-11
 - Delete Session, 6-12
 - Swap Session, 6-12
 - parameters, 6-10
 - purpose, 6-10
 - usage, 6-11
- swap session, 6-12
- Swap / Display User Sessions screen, 6-11
- &symbolic name parameter, description, with SUBMIT command, 7-10
- symbolic substitution, in commands, 2-6
- synchronized submit, in batch interface, 3-5
- syntax, batch interface, 3-2
- SYSIN, ddname in batch job, 3-3

T

- TABLE parameter, general use, 2-11
- TABLE parameter, description
 - with SELECT PROCESS command, 8-18
 - with SELECT STATISTICS command, 9-12
- TCPNAME, SNODE subparameter, SUBMIT command, 7-7
- TCQ (Transmission Control Queue), description of, 1-6
- Temporary file, when it is used, 2-11
- Transmission Control Queue (TCQ). *See* TCQ
- TSO, IUI under, 1-2
- tutorial, 4-7
- TYPE parameter, description, with SELECT STATISTICS command, 9-10

U

- underlined letters, notational conventions, xvi
- upper case, print control with SIGNON, 6-2
- uppercase and lowercase letters, notational conventions, xvi
- uppercase letters, notational conventions, xvi
- User Authorization screen, 4-6
- User Information screen (WHO), 6-6
- user notification programs, 12-1
- user notify program, DMNOTFY2, 12-3
- USER parameter, description, with SELECT STATISTICS command, 9-10
- USERID
 - parameter with SIGNON, 6-4
 - with RECONNECT parameter, 6-3
- using the DMRTDYN sample program, 12-6
- Utility Programs
 - DMRTPOMV, 12-7
 - DMRTPOVM, 12-7
- utility programs
 - DMNOTFY2, 12-3
 - DMNOTIFY, 12-2
 - DMRTDYN, 12-4

V

- vertical bars, notational conventions, xvi
- viewing
 - Message Text, 4-12
 - the Message display, 4-13
- VTAM, re-establish session, 6-3

W

- WHERE parameter, general use, 2-10
- WHERE parameter, description
 - with CHANGE PROCESS command, 8-3
 - with DELETE PROCESS command, 8-11
 - with FLUSH PROCESS command, 8-11
 - with SELECT NETMAP command, 10-2
 - with SELECT PROCESS command, 8-16
 - with SELECT STATISTICS command, 9-4
 - with SUSPEND PROCESS command, 8-11
- WHO option, 6-6

X

- X, use as IUI SIGNOFF , 6-14

Y

- year 2000 compliance, 7-8, 9-6, 9-8