

Connect:Direct[®] for VSE/ESA

Administration Guide

Version 3.2

**Connect:Direct for VSE/ESA Administration Guide
Version 3.2**

Third Edition

(c) Copyright 1986-2006 Sterling Commerce, Inc. All rights reserved. Additional copyright information is located at the end of the release notes.

STERLING COMMERCE SOFTWARE

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

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

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

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

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

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

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

Sterling Commerce, Inc.
4600 Lakehurst Court Dublin, OH 43016-2000 *
614/793-7000

Contents

Chapter 1 About Connect:Direct for VSE/ESA	11
Connect:Direct for VSE/ESA Internal Components	11
Chapter 2 Using Administration Commands	13
Using Connect:Direct Tasks	13
Entering Task Commands	14
Removing Tasks from Execution	15
Command Format	15
Required Parameter	15
Optional Parameter	15
Using FLUSH TASK Through Batch Interface	15
Displaying Tasks	16
Command Format	16
Required Parameters	16
Optional Parameters	16
Using SELECT TASK Through Batch Interface	16
Displaying Initialization Parameter Settings	16
Command Format	16
Required Parameters	17
Optional Parameters	17
Using INQUIRE INITPARMS Through Batch Interface	17
Displaying Statistics Logging Facility Status	20
Command Format	20
Required Parameters	20
Optional Parameters	20
Using INQUIRE STATISTICS Through Batch Interface	20
Displaying Statistics Archive File Directory	21
Command Format	21
Required Parameters	22
Optional Parameters	22
Using INQUIRE STATDIR Through Batch Interface	23
Switching Statistics File Pair	24
Command Format	24
Required Parameters	24
Optional Parameters	24
Using STATISTICS SWITCH Through Batch Interface	24

Recording Statistics for Specific Record Types	25
Command Format	25
Required Parameters	25
Optional Parameters	25
Using STATISTICS ON/OFF Through Batch Interface	25
Notifying Connect:Direct of Statistics File Archival	26
Command Format	26
Required Parameters	26
Optional Parameters	27
Using STATISTICS ARCHIVED Through Batch Interface	27
Running Connect:Direct Diagnostics	27
Command Format	27
Required Parameters	28
Optional Parameters	28
Using MODIFY Through Batch Interface	28
Stopping Connect:Direct for VSE/ESA	29
Command Format	29
Required Parameters	29
Optional Parameters	29
Using STOP CD Through Batch Interface	30

Chapter 3 Using Type Key File Commands 31

Description of Type Key Defaults File	31
Overriding File Attributes	31
MS-DOS and OS/2 Type Keys	32
TEXT Type Key	32
DF Type Key	32
DF2 Type Key	32
BINARY Type Key	33
Maintaining the Type File	33
Using INSERT TYPE Command	33
Command Format	33
Required Parameter	34
Optional Parameters	34
Using INSERT TYPE Through Batch Interface	37
Using UPDATE TYPE Command	37
Command Format	38
Required Parameter	38
Optional Parameters	38
Using UPDATE TYPE Through Batch Interface	41
Using DELETE TYPE Command	42
Command Format	42
Required Parameter	42
Optional Parameters	42
Using DELETE TYPE Through Batch Interface	43
Using SELECT TYPE Command	43
Command Format	43
Required Parameters	44
Optional Parameters	44
Using SELECT TYPE Through Batch Interface	44

Chapter 4 Maintaining User Authorization File 45

Authorization Facility	45
Authorization File	45
Maintaining the User Authorization File	45
Using INSERT USER Command	46
Command Format	46
Required Parameters	48
Optional Parameters	48
Authorization Record Parameters	48
Functional Authorization Parameters	49
Using INSERT USER Through Batch Interface	50
Using UPDATE USER Command	51
Command Format	51
Required Parameters	53
Optional Parameters	53
Authorization Record Parameters	53
Functional Authorization Parameters	54
Using UPDATE USER Through Batch Interface	55
Using DELETE USER Command	56
Command Format	56
Required Parameter	57
Optional Parameters	57
Using DELETE USER Through Batch Interface	57
Using SELECT USER Command	58
Command Format	58
Required Parameters	58
Optional Parameters	59
Using SELECT USER Through Batch Interface	59

Chapter 5 Defining and Maintaining the Network Map 61

Definitions	61
Local Node Record	61
Adjacent Node Record	62
Defining the Local Node as an Adjacent Node	62
Examples	62
Keywords	64
LOCAL.NODE Keyword	64
Positional Parameters	64
Keyword Parameter	65
Defining Local Node as Adjacent Node	65
ADJACENT.NODE Keyword	66
Positional Parameters	66
Keyword Parameters	69
Sample Connect:Direct Network Map Entries	71
PNODE=SNODE Processing	72
Example Node Records	72
Local Node and Adjacent Node Record for the Local Node	72
VSE or MVS SNA LU0 Adjacent Node Example	72

VSE or MVS TCP/IP Adjacent Node Examples	73
VSE or MVS SNA LU6.2 Adjacent Node Example	73
VSE or MVS Trusted Node Example	73
OpenVMS LU0 Adjacent Node Example	74
MS-DOS LU0 Adjacent Node Examples	74
MS-DOS LU6.2 Adjacent Node Example	74
NetWare LU6.2 Adjacent Node with Independent LU Name Designated	74
NetWare LU6.2 Adjacent Node with Dependent LUs	75
NetWare LU6.2 Adjacent Node with Independent and Dependent LUs	75
NetWare TCP/IP Adjacent Node Example	75
OS/2 LU0 Adjacent Node Examples	76
OS/2 LU6.2 Adjacent Node with Independent LU Example	76
OS/2 LU6.2 Adjacent Node with Dependent LU Example	76
OS/400 SNUF Adjacent Node Example	76
OS/400 (LU6.2) Adjacent Node with Independent LU Example	77
OS/400 (LU6.2) Adjacent Node with Dependent LU Example	77
Tandem LU0 Adjacent Node Example	77
UNIX TCP/IP Adjacent Node Example	77
UNIX LU6.2 Adjacent Node Example	78
VM SNA LU0 Adjacent Node Example	78
VSE SNA LU0 Adjacent Node Example	78
VSE LU6.2 Adjacent Node Example	79
Windows NT TCP/IP Adjacent Node Example	79
Updating the Network Map	79
Updating While Connect:Direct is Not Executing.	80
Updating the Network Map Dynamically	80
Using \$\$ACTION VERBS	81
Basic Action Verbs	81
Special Purpose Verb Pairs	82
Update Using the Connect:Direct Batch Utility	82
\$\$ACTION Verb Examples	83
\$\$INSERT Example	83
\$\$UPDATE Example	83
\$\$DELETE Example	84
\$\$SYNTAX Example	85
\$\$VERIFY Example	86

Chapter 6 Planning and Controlling Security 87

Planning for Security	87
Levels of Security Support	87
SECURITY Initialization Parameter	88
Authorization Facility	88
Security Exits Invoked During Processing	88
Security During Signon Command	89
Security During Process Execution	91
Implementing Security Exits	92
Stage 1 Signon Security Exit	92
Stage 2 Security Exit	92
Run Job Security Exit	93
Connect:Direct Authorization Facility	93
Example of Authorization Facility Use	93

Chapter 7 Using Optional Connect:Direct for VSE/ESA Exits **95**

Statistics Exit	95
Calling Conventions of the Statistics Exit	96
ESA Systems	96
Statistics Records	97
Connect:Direct for VSE/ESA Statistics Records	97
Statistics Records	98
Submit Exit	100
Submit Exit Processing Flow	100
Implementation of Submit Exit	100
Stage 1 Submit Exit	101
Stage 2 Submit Exit	101
Control Block Format	101
Example of Created Control Block	102
Modifiable TCQE Fields	103
Conversion of Parallel Session Values	105
Allocation Exit	106
How the Allocation Exit Executes	106
Calculating Addresses and Values	107
Copy Control Block Definitions	108
Copy Control Block Modifications	108
DDESCR Control Block Format	109
DDESCR Modifiable Fields	114
I/O Exit	116
Implementing the I/O Exit	116
Specifying the I/O Exit on the COPY Statement	116
Specifying the I/O Exit in the TYPE File	117
I/O Exit Access to Control Blocks	117
I/O Exit Requests	119
Normal Input Calling Sequence	121
Normal Output Calling Sequence	121

Chapter 8 Maintaining Connect:Direct for VSE/ESA **123**

Step 1 – Install Maintenance Tapes	123
Step 2 – Maintain Network Map Definitions	124
Step 3 – Add Messages to Connect:Direct Message File	124
Sample Format for Message Source	124
Rules for Variables and Message IDs	124
Jobstream to Update Message File	125

Chapter 9 Using Statistics Administration **127**

Statistics Facility Overview	127
File Pairs	127
Order	127
Switching	128
Archiving Statistics	128

Statistics Archive File Directory	130
Archiving Related Utilities	130
DMSTARRT	130
Sample Archiving Process Using DMSTARRT	131
DMSTARBT	131
Sample Archiving Process Using DMSTARBT	132
DMSTBKEY	132
Sample Jobstream to Execute DMSTBKEY	133
Sample Archiving Setup	133
Configuring the Statistics Facility	133
Monitoring the Statistics Facility	136
INQUIRE STATISTICS Command	137
S2 Statistics Records	137
SCCSTAT	138
SS Statistics Records	138
Changing the File Pair Configuration	139
File Pair Change Restrictions	139
File Pair List Changes	139

Chapter 10 Programming Interface to Connect:Direct for VSE/ESA 141

Interfacing with Connect:Direct	141
Application Interface Program	141
Connect:Direct Command Strings	141
Writing a User Application Program	142
API High-Level Interface (DMCHLAPI)	142
Completing the Required Parameters	143
Using UICB Extract Feature	144
UICB Field Listing	146
Control Block Builder Syntax Error Work Areas	147
Start of Boolean Flags	148
How DMCHLAPI Interprets Parameters	150
DMCHLAPI Return Codes	151
Extract Function Fields of Interest	151
Examples of DMCHLAPI Parameters and Calling Sequences	152
DMCHLAPI Parameters for an Assembler Program	152
DMCHLAPI Parameters for a PL/I Program	152

Chapter 11 About the Connect:Direct CICS Interface Administration System 153

Introduction to the Administration System	155
Structure of the Administration System	156
Administration Screen Features	157
System Fields	157
PF Keys	157

Chapter 12 Using the Connect:Direct CICS Interface Administration System 159

Signing On and Off	159
Using the Primary Menu	159
Entry Fields	160
Understanding the Administration Interface	161
Operating System Subtasks	161
Monitor Transaction	161
Interface Tasks	162
Interface Startup	162
Interface Subtask Management	162
Interface Request Management	162
Interface VTAM Session	163
Using the Administration Interface Screen	163
Operational Considerations	165
Signing On to One DTF from Multiple Connect:Direct CICS Interface Facilities	165
Shutting Down CICS	166
Restart Considerations for Task ABENDS	166
Accounting and Logging Information	167
ESF Operation	167
Connect:Direct SIGNON Parameters	167
Connect:Direct CICS Interface Data Sets	168
Connect:Direct DTF Data Sets	168
Connect:Direct CICS Interface Data Sets	169

Chapter 13 Configuring the Connect:Direct CICS Interface Administration System 171

Entry Fields	171
Configuring the Control Record	172
Entry Fields	173
Menu Options	175
How to Change the Control Record Information	176
Configuring the DTF Node Records	176
How to Update the DTF Node Records Information	176
Adding a DTF Node Record	177
Deleting a DTF Node Record	177
Reading a DTF Node Record	177
Editing a DTF Node Record	177
DTF Node Records Screen	177
Entry Fields	178
Configuring Network Node Records	179
How to Update the Network Node Records Screen	179
Adding a Network Node Record	179
Deleting a Network Node Record	180
Reading a Network Node Record	180
Editing a Network Node Record	180
Network Node Records Screen	180
Entry Fields	181

Chapter 14 Monitoring Status with the Connect:Direct CICS Interface Administration System 183

Node Status and Control	183
Line Commands	183
System Fields	184
How to Control Accessible Nodes	185
Viewing the Work Queue	185
System Fields	186
How to View the Work Queue Display	186
Viewing User Status	186
Options	187
System Fields	187
Monitoring User Status	188

Chapter 15 Using Connect:Direct CICS Interface Signon Defaults 191

Entry Fields	192
System Fields	193
Updating the Signon Defaults Screen	193
Adding a User Sign-On Record	193
Deleting a User Sign-On Record	194
Reading a User Sign-On Record	194
Editing a User Sign-On Record	194

Chapter 16 Using the Connect:Direct CICS Interface Application Program Interface 195

Driver Fields	196
Connect:Direct CICS Interface API Sample Program	196

Appendix A Connect:Direct CICS Interface Administration Messages 199

Appendix B DBCS Support 203

Understanding DBCS	203
DBCS Representation	203
Default Translation Tables	204
Customizing the Translation Tables	204
Required Parameters	205
Optional Parameters	206
Comments	211
Using the RULES Parameter	211
Using the SBCS Parameter	212
Using the DBCS Parameter	212
Sample Preprocessor Input Data Stream	213
Sample JCL to EXECUTE the Preprocessor	213

About Connect:Direct for VSE/ESA

Connect:Direct links technologies and moves all types of information between networked systems/computers. It manages high-performance transfers by providing such features as: automation, reliability, efficient use of resources, application integration, and ease of use. Connect:Direct offers choices in communications protocols, hardware platforms, and operating systems. It provides the flexibility to move information among mainframes, midrange systems, desktop systems, and LAN-based workstations.

Connect:Direct for VSE/ESA Internal Components

Connect:Direct for VSE/ESA has three internal components:

- ◆ Application program interface (API)
- ◆ Data transmission facility (DTF)
- ◆ Various user interfaces to Connect:Direct for VSE/ESA

Connect:Direct for VSE/ESA runs as an application on the VSE/ESA operating system. Within Connect:Direct for VSE/ESA the system components interact to execute the statements and commands submitted through the Connect:Direct interfaces. All statements and commands pass through the API, regardless of the interface from which they are submitted.

Using Administration Commands

Connect:Direct administration commands allow you to complete the following administration functions:

- ◆ Using Connect:Direct Tasks
- ◆ Removing Tasks from Execution
- ◆ Displaying Tasks
- ◆ Displaying Initialization Parameter Settings
- ◆ Displaying Statistics Logging Facility Status
- ◆ Displaying Statistics Archive File Directory
- ◆ Switching Statistics File Pair
- ◆ Recording Statistics for Specific Record Types
- ◆ Notifying Connect:Direct of Statistics File Archival
- ◆ Running Connect:Direct Diagnostics
- ◆ Stopping Connect:Direct for VSE/ESA

Using Connect:Direct Tasks

Connect:Direct tasks are service functions or requests processed by the DTF. The tasks are represented by a task control area (TCA) which is used by the DTF to process and manage the request.

The Connect:Direct tasks can be divided into two categories, system tasks and user tasks. System tasks perform the services needed for the operation of the Connect:Direct DTF. User tasks represent work done within the DTF on behalf of a user request.

The following table lists the Connect:Direct tasks and their functions.

Type	Task	Function
System	Master (M)	Controls the dispatching and logon processing for the DTF.
	Fitscan (F)	Controls initiating and dispatching new tasks.
	Timer (T)	Performs timer services for the master task and process related timer functions.
	Operator interface (O)	Allows the user to communicate to the DTF through the operator console.
	Extended Submit Facility Scan (W)	Scans the TCQ at predefined intervals and moves processes submitted that are not on the current processing queue (PCQ) to the PCQ.
	TCP Task (U)	Listens for incoming TCP/IP session requests.
	LOGON (L)	Reserved for use during logon processing.
User	PNODE Task (P)	Created for managing the work related to a request that initiated the current session.
	SNODE Task (S)	Created for managing the work related to a partner PNODE Task.
	IUI Task (I)	Created by the DTF to manage the requests from a session with an IUI user.

Entering Task Commands

The two task commands can be entered through all interfaces:

Command	Description
FLUSH TASK	removes a Process or a signed-on user from the

Connect:Direct for VSE/ESA system based on the physical system task being used to execute it.

Command	Description
SELECT TASK	selects and displays the list of all active tasks.

Removing Tasks from Execution

The FLUSH TASK command removes a task from execution by task number. A Process can be flushed using this command, but should only be used if the Process cannot be flushed using the Flush Process command.

Command Format

The FLUSH TASK command has the following format and associated parameters.

Label	Command	Parameters
(optional)	FLUSH TASK	WHERE (TASK = tasknumber (list)) FORCE

Required Parameter

The following is the required parameter for the FLUSH TASK command.

Parameter	Description
WHERE TASK = tasknumber (list)	specifies which tasks to flush and the method to use. tasknumber (list) specifies, either by task number or a list of task numbers, the task or tasks to flush.

Optional Parameter

The FORCE parameter is optional for the FLUSH TASK command.

Parameter	Description
FORCE	specifies that the task flush will be forced. Note: Do not use the FORCE parameter when the task is executing on a LU6.2 session. If FORCE is specified, the session will terminate immediately and statistics for the task will not be exchanged between the two nodes.

Using FLUSH TASK Through Batch Interface

To use the FLUSH TASK command from the batch interface, place your commands in a batch job stream as described in Step 10 of the installation procedure, and submit the job while Connect:Direct is running.

The following example illustrates the use of the FLUSH TASK command to force flush three Connect:Direct tasks:

```
FLUSH TASK WHERE (TASK=(100,105,120) FORCE)
```

Displaying Tasks

The SELECT TASK command selects and displays the status of Connect:Direct system tasks.

Command Format

The SELECT TASK command has the following format and associated parameters.

Label	Command	Parameters
(optional)	SElect TASK	(none)

Required Parameters

There are no required parameters for the SELECT TASK command.

Optional Parameters

There are no optional parameters for the SELECT TASK command.

Using SELECT TASK Through Batch Interface

To use the SELECT TASK command from the batch interface, place your commands in a batch job stream like the example presented in the *Utility Programs* chapter of the *Connect:Direct for VSE/ESA User's Guide*, and submit the job while Connect:Direct is running. The following example illustrates the use of SELECT TASK command:

```
SEL TASK
```

Displaying Initialization Parameter Settings

The INQUIRE INITPARM command displays the current settings of the DTF initialization parameters.

Command Format

The INQUIRE INITPARMS command has the following format.

Label	Command	Parameters
(optional)	INQUIRE INITparm	

Required Parameters

There are no required parameters for the INQUIRE INITPARM command.

Optional Parameters

There are no optional parameters for the INQUIRE INITPARM command.

Using INQUIRE INITPARMS Through Batch Interface

To use the INQUIRE INITPARM command from the batch interface, place your commands in a batch job stream as shown in the description of the batch interface program in the *Using Utility Commands* chapter of the *Connect:Direct for VSE/ESA User's Guide*, and submit the job while Connect:Direct is running. You must set the fifth character of the DMBATCH output parameter specification to a Y to print the result of the command that is in the temporary data set.

A report showing the current settings of the DTF initialization parameters will display. The following figure shows a partial sample report.

```

=====
TEST32.VSE32      *INQUIRE INITPARM*  DATE: 08.26.1997  TIME: 07:36:36
=====

ALLOC.CODES       => (0068 0069 006A 020C 0210 0218 0220 0234)
ALLOC.RETRIES     => 20
ALLOC.STORAGE     => ABOVE
ALLOC.WAIT        => 00:03:00
ALLOCATION.EXIT    =>
BUFND             => 6
CKPT              => 0
CKPT.DAYS         => 4
CKPT.MODE         => (RECORD BLOCK PDS NOPDS VSAM VSAM)
CONNECT.WAIT      => 3
DATEFORM         => MDY
DEBUG             => '80000080'
DYNT.AUTOCAT     => NO
DYNT.DYNAMT      => NO
ESF.WAIT          => 00:03:00
ESTAE             => YES
LOG.PRINTER      =>
LU2.WAIT         => 30
MAXBATCH         => 0
MAXPRIMARY       => 6
MAXPROCESS       => 12
MAXRETRIES       => 7
MAXSECONDARY     => 6
MAXSTGIO         => 61440
MAXUSERS         => 6
MCS.SIGNON       => SIGNON (DISPLAY SUPPRESSED)
NDM.KEY          => 85WTXXZ6ANMZ9P
NDM.NODE         => 1
NETMAP.CHECK     => ALL,ALL,FAIL
PC.ENABLE.CHECK  => No
POWER.LST.CLASS  => ABCD
POWER.LST.NETMAP =>
POWER.LST.PROC   =>
POWER.LST.SNODE  =>
POWER.LST.USERID1 => (LSTCD1,,,)
POWER.LST.USERID2 => (LSTCD222,,,)
POWER.MPWD       => .....
POWER.PUN.CLASS  => WXYZ
POWER.PUN.NETMAP =>
POWER.PUN.PROC   =>
POWER.PUN.SNODE  =>
POWER.PUN.USERID1 => (PUNCD1,PUNPROC,SS.DUB.TEST31,
                       TEST.NETMAP.FILE)
POWER.PUN.USERID2 => (PUNCD222,,,)
POWER.PUN.USERID3 => (PUNCD333,,,
                       TEST.NETMAP.FILE)
PRTYDEF         => 10
QUIESCE         => No
REUSE.SSESSIONS => Yes
RUN.JOB.EXIT    =>
RUN.TASK.EXIT    =>
SECURITY.EXIT    => OFF
STAT.ARCH.CONFIRM => No
STAT.ARCH.DIR   =>

```

(continued)

STAT.BUFFER.ESDSDATA	=> 6
STAT.BUFFER.KSDSDATA	=> 6
STAT.BUFFER.KSDSINDX	=> 6
STAT.DSN.BASE	=> TEST31.STATS
STAT.ERROR	=> Disable
STAT.EXCLUDE	=>
STAT.FILE.PAIRS	=> 2
STAT.INIT	=> Warm
STAT.QUEUE.ELEMENTS	=> 99
STAT.SWITCH.SUBMIT	=> N(ARCVSETP)
STAT.SWITCH.TIME	=>
STAT.USER	=>
STATISTICS.EXIT	=>
STAT.TPREC	=> (00:00:00,24:00:00,00)
STRNO.AUTH	=> 13
STRNO.CKPT	=> 13
STRNO.DEFAULT	=> 4
STRNO.MSG	=> 10
STRNO.NETMAP	=> 4
STRNO.STAT	=> 10
STRNO.TCQ	=> 6
STRNO.TCX	=> 3
STRNO.TYPE	=> 13
SUBMIT.EXIT	=>
TCP	=> NO
TCP.BUFSIZE	=> 4096
TCP.NAME	=> 00
TCP.PORTNUM	=>
TCP.TIMER	=> 0
TCQ	=> COLD
UPPER.CASE	=> No
V2.BUFSIZE	=> 4096
WTMESSAGE	=> NO
WTRETRIES	=> 00:03:00
ECZ.COMPRESSION.LEVEL	=> 1
ECZ.MEMORY.LEVEL	=> 4
ECZ.WINDOWSIZE	=> 13
AUTHDSN	=> TEST31.AUTH
CKPTDSN	=> TEST31.CKPT
LU1.SCRIPT.DSN	=>
MSGDSN	=> TEST31.MSG
NETDSN	=> TEST31.NETMAP
TCQDSN	=> TEST31.TCQ
TCXDSN	=> TEST31.TCX
TYPEDSN	=> TEST31.TYPE
AUTHCAT	=>
CKPTCAT	=>
MSGCAT	=>
NETCAT	=>
STATCAT	=>
TCQCAT	=>
TCXCAT	=>
TYPECAT	=>

Displaying Statistics Logging Facility Status

The INQUIRE STATISTICS command displays the current status of the Connect:Direct statistics logging facility.

Command Format

The INQUIRE STATISTICS command has the following format.

Label	Command	Parameters
(optional)	INQUIRE STATISTICS	

Required Parameters

There are no required parameters for the INQUIRE STATISTICS command.

Optional Parameters

There are no optional parameters for the INQUIRE STATISTICS command.

Using INQUIRE STATISTICS Through Batch Interface

To use the INQUIRE STATISTICS command from the batch interface, place your commands in a batch job stream as shown in the description of the batch interface program in the *Using Utility Commands* chapter of the *Connect:Direct for VSE/ESA User's Guide* and submit the job while Connect:Direct is running. You must set the fifth character of the DMBATCH output parameter specification to a **Y** to print the result of the command that is in the temporary data set.

A report showing the current status of the statistics logging facility will display. The report will include information such as the configuration of the statistics file pair list, the active file pair, file percentage utilizations, date and time ranges in the files, and additional information about the statistics facility. The following figure shows a sample report.

```

=====
node name      *INQ STATS* DATE: mm/dd/yyyy TIME: hh:mm:ss
=====

Status        => Enabled                Sec. Name    => USER01
Return Code   => 0                      Message ID   => SSTL000I
Last "S2"     => 00:00:00              Que Wait    => No
Dsn Base      => USER01.STTX
Excluded      => MC
*****      F I L E P A I R #01      *****
Status        => Active
Start Date    => 03/20/1997            End Date     => 03/20/1997
Start Time    => 15:25:29              End Time     => 15:51:35
KSDS Size     => 196608                ESDS CIS     => 4096
ESDS Size     => 2211840              ESDS Loc.    => 1474
ESDS Util.    => 1%
Reset Pend.   => No                    Arch. Wait   => No
Last Switch   =>                      Sel. Count   => 0
KSDS Status   => Alloc, Open            ESDS Stat.   => Alloc, Open
  L-cmd       => ENDREQ                 L-cmd       => ENDREQ
  L-msg       => SVS0000I              L-msg       => SVS0000I
  L-rc        => 0                      L-rc        => 0
  L-fdb       => 0                      L-fdb       => 0
*****      F I L E P A I R #02      *****
Status        =>
Start Date    =>                      End Date     =>
Start Time    =>                      End Time     =>
KSDS Size     => 196608                ESDS CIS     => 4096
ESDS Size     => 2211840
ESDS Util.    => 0%                    KSDS Util.   => 0%
Reset Pend.   => No                    Arch. Wait   => No
Last Switch   =>                      Sel. Count   => 0
KSDS Status   => Alloc                ESDS Stat.   => Alloc
  L-cmd       =>                      L-cmd       =>
  L-msg       =>                      L-msg       =>
  L-rc        => 0                      L-rc        => 0
  L-fdb       => 0                      L-fdb       => 0

```

Displaying Statistics Archive File Directory

The INQUIRE STATDIR command displays the Connect:Direct statistics archive file directory.

Command Format

The INQUIRE STATDIR command has the following format and associated parameters.

Label	Command	Parameters
(optional)	INQUIRE STATDIR	STARTT = ([date day] [,hh:mm:ssXM])

Required Parameters

There are no required parameters for the INQUIRE STATDIR command.

Optional Parameters

The following are descriptions for the optional parameters used with the INQUIRE STATDIR command.

Parameter	Description
STARTT = ([date day] [,hh:mm:ssXM])	<p>specifies that directory display is to begin with the first archive file created after the designated starting date and time. The date or day and time are positional parameters. If you do not specify the date or day, a comma must precede the time. If you omit this parameter, the display will begin with the first directory entry.</p> <p>date specifies the date from which the statistics records will be selected. You can specify the day (dd), month (mm), and year (yy).</p> <p>Connect:Direct for VSE/ESA is Year 2000 compliant and detailed information on Year 2000 compliance is presented in the <i>Connect:Direct for VSE/ESA Release Notes</i>.</p> <p>To specify the order of a Gregorian day, month, and year, you <i>must</i> define the DATEFORM initialization parameter. If you do not specify the DATEFORM parameter, Connect:Direct for VSE/ESA defaults to the VSE/ESA user setup option STD OPT DATE which is specified at system initial program load (IPL).</p> <p>Once you have specified the order, you can use the following formats according to the order you selected:</p> <p><u>DATEFORM=MDY</u> accepts the following formats: mmddy or mmddy mm/dd/yy or mm/dd/yyyy mm.dd.yy or mm.dd.yyyy</p> <p><u>DATEFORM=DMY</u> accepts the following formats: ddmmy or ddmmyy dd/mm/yy or dd/mm/yyyy dd.mm.yy or dd.mm.yyyy</p> <p><u>DATEFORM=YMD</u> accepts the following formats: yyymm or yyymm yy/mm/dd or yyyy/mm/dd yy.mm.dd or yyyy.mm.dd</p> <p><u>DATEFORM=YDM</u> accepts the following formats: yyddm or yyddm yy/dd/mm or yyyy/dd/mm yy.dd.mm or yyyy.dd.mm</p>

Parameter	Description
STARTT (<i>continued</i>)	<p>Connect:Direct for VSE/ESA also processes Julian dates the same as previous Connect:Direct for VSE releases. The following formats are valid:</p> <p>yyddd or yyyyddd yy/ddd or yyyy/ddd yy.ddd or yyyy.ddd</p> <p>If you only specify date, the time will default to 00:00.</p> <p>day specifies to display the first archive file created after this day of the week. Valid names include MOnday, TUEsday, WEdnesday, THursday, FRiday, SATurday, and SUNday. You can also specify YESTER to search for archive files created after yesterday or TODAY to search for the archive files created after today.</p> <p>hh:mm:ssXM indicates to display the first archive file created after this time of day, specified in hours (hh), minutes (mm), and seconds (ss). 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.</p> <p>If you do not use either AM or PM, Connect:Direct assumes the 24-hour clock. You do not need to specify minutes and seconds. You can also specify NOON, which will display files created after noon, or MIDNIGHT, which will display archive files created after midnight. The default for the time is 00:00:00, the beginning of the day.</p> <p>If you specify time of day but not date, the output will show the first available entry in the archive directory for files created after that time of day. Archive files from all later times and dates will display up to and including the stop time.</p> <p><u>Default:</u> 00:00:00</p>

Using INQUIRE STATDIR Through Batch Interface

To use the INQUIRE STATDIR command from the batch interface, place your commands in a batch job stream as shown in the description of the batch interface program in the *Using Utility Commands* chapter of the *Connect:Direct for VSE/ESA User's Guide* and submit the job while Connect:Direct is running. You must set the fifth character of the DMBATCH output parameter specification to a **Y** to print the result of the command that is in the temporary data set.

Supply the beginning date and time to limit the display for the INQUIRE STATDIR command. A report showing the results of the inquiry will display.

The following figure shows a sample report.

```

=====
node.name          *INQUIRE STATDIR*  DATE:  mm/dd/yyyy  TIME:  hh:mm:ss
=====

Archival DSN:      USER01.STT.ARCHSTAT.G0008V00
Archival Notification: 03/02/97  97.061  00:01:28
Oldest Record:    03/01/97  97.060  00:00:06
Newest Record:    03/01/97  97.060  23:59:54

Archival DSN:      USER01.STT.ARCHSTAT.G0009V00
Archival Notification: 03/03/97  97.062  00:01:35
Oldest Record:    03/02/97  97.061  00:00:11
Newest Record:    03/02/97  97.061  23:59:45

Archival DSN:      USER01.STT.ARCHSTAT.G0010V00
Archival Notification: 03/04/97  97.063  00:01:53
Oldest Record:    03/03/97  97.062  00:00:08
Newest Record:    03/03/97  97.062  23:59:41

```

Switching Statistics File Pair

The STATISTICS SWITCH command initiates a statistics file pair switch. The currently active file pair will close, and logging will continue on the next file pair in sequence. This command provides a means of initiating a file pair switch at any given time. Otherwise, switching occurs when the active file pair fills, or when a time of day specified in the STAT.SWITCH.TIME initialization parameter occurs.

Command Format

The STATISTICS SWITCH command has the following format.

Label	Command	Parameters
(optional)	STATistics	SWITCH

Required Parameters

There are no required parameters for the STATISTICS SWITCH command.

Optional Parameters

There are no optional parameters for the STATISTICS SWITCH command.

Using STATISTICS SWITCH Through Batch Interface

To use the STATISTICS SWITCH command from the batch interface, place your commands in a batch job stream as shown in the description of the batch interface program in the *Using Utility*

Commands chapter of the *Connect:Direct for VSE/ESA User's Guide* and submit the job while Connect:Direct is running.

Recording Statistics for Specific Record Types

The STATISTICS ON/OFF command enables or disables recording of specific statistics record types. When you initialize the DTF, Connect:Direct enables the recording of all record types unless you specify the STAT.EXCLUDE initialization parameter. You can use the INQUIRE STATISTICS command to find out which types are currently disabled.

Command Format

The STATISTICS ON/OFF command has the following format and associated parameters.

Label	Command	Parameters
(optional)	STATistics ON OFF	TYPE = (record type list)

Required Parameters

The following parameter is required for the STATISTICS ON/OFF command.

Parameter	Description
TYPE	Specifies the list of statistics record types whose recording is to be enabled or disabled. Use the two-character identifier to specify record types. A list of these identifiers is given in <i>Statistics Exit</i> on page 95.

Optional Parameters

There are no optional parameters for the STATISTICS ON/OFF command.

Using STATISTICS ON/OFF Through Batch Interface

To use the STATISTICS ON/OFF command from the batch interface, place your commands in a batch job stream as shown in the description of the batch interface program in the *Using Utility Commands* chapter of the *Connect:Direct for VSE/ESA User's Guide* and submit the job while Connect:Direct is running.

The use of the STATISTICS ON/OFF command to exclude statistics record logging should be done prudently, because some types of records are critical for problem diagnosis. Do not exclude the following record types:

- ◆ CT Copy Termination
- ◆ PS Process Submit

- ◆ PT Process Termination
- ◆ RJ Run Job
- ◆ RT Run Task
- ◆ SW Submit within Process
- ◆ WO WTO

Other record types are usually less critical and can be excluded.

Note: Excluding record types can make problem analysis and resolution more difficult.

Notifying Connect:Direct of Statistics File Archival

The STATISTICS ARCHIVED command notifies Connect:Direct that the indicated statistics file has been archived. This notification allows the system to erase and overwrite the file with new records. When you specify STAT.ARCH.CONFIRM=YES in the DTF initialization parameters, Connect:Direct cannot reuse a statistics file pair until receiving confirmation that the archive is complete. This command provides an additional means of sending this notification. Ordinarily it is sent by the DMSTARRT utility after the archive is done by a Connect:Direct COPY Process, or by the DMSTARBT utility after the archive is done by a batch step. Refer to *Archiving Statistics* on page 128.

Command Format

The STATISTICS ARCHIVED command has the following format and associated parameters.

Label	Command	Parameters
(optional)	STATistics ARCHived	file pair number

Required Parameters

The following parameter is required for the STATISTICS ARCHIVED command.

Parameter	Description
file pair number	Specifies a number from 1-20 that identifies the statistics file for which archive notification is to be sent. This is given as the relative number of the file pair in the file pair list. The first pair in the list is file pair number 1.

Optional Parameters

There are no optional parameters for the STATISTICS ARCHIVED command.

Using STATISTICS ARCHIVED Through Batch Interface

To use the STATISTICS ARCHIVED command from the batch interface, place your commands in a batch job stream as shown in the description of the batch interface program in the *Using Utility Commands* chapter of the *Connect:Direct for VSE/ESA User's Guide* and submit the job while Connect:Direct is running.

Running Connect:Direct Diagnostics

The MODIFY command allows you to run Connect:Direct diagnostics. You can dynamically modify operational functions and request traces. The MODIFY command yields the same types of traces as the DEBUG initialization parameter, except that MODIFY does not require bringing down and restarting Connect:Direct. Normally, the trace begins immediately after the MODIFY command is issued.

Note: The correct DLBL statements must be present in the start-up JCL for tracing to capture data.

Command Format

The MODIFY command has the following format and associated parameters:

Label	Command	Parameters
(optional)	MODIFY	BITS.OFF = X'nnnnnnnn' BITS.ON = X'nnnnnnnn'

You can issue the MODIFY command through batch interface or operator interface. Refer to the *Connect:Direct Console Operator's Guide* for information on using the operator interface.

Required Parameters

The following are descriptions for the required parameters used with the MODIFY command.

Parameter	Value																																																						
BITS.OFF=X'nnnnnnnn'	The following debug settings specify valid values of nnnnnnnn:																																																						
BITS.ON = X'nnnnnnnn'	<table border="1"> <thead> <tr> <th>DEBUG Setting</th> <th>Trace Type</th> <th>Output DLBL Name</th> </tr> </thead> <tbody> <tr> <td>80000000</td> <td>COPY Routine and RUN TASK trace</td> <td>RADB01</td> </tr> <tr> <td>20000000</td> <td>Display of final TCQE from DMCBSUBM</td> <td>TCQE</td> </tr> <tr> <td>10000000</td> <td>Full TPCB/SYMBOLICS from DMCBSUB</td> <td>DMCBSUB</td> </tr> <tr> <td>08000000</td> <td>Session manager trace.</td> <td>RADB05</td> </tr> <tr> <td>04000000</td> <td>Separate trace per task. Example: "R0000005" to trace TASK 5</td> <td>Rnnnnnn</td> </tr> <tr> <td>02000000</td> <td>API session trace</td> <td>RADB07</td> </tr> <tr> <td>01000000</td> <td>DMGCSUB trace</td> <td>RADB08</td> </tr> <tr> <td>00400000</td> <td>TCQSH from DMCBCOPY</td> <td>DMCBCOP</td> </tr> <tr> <td>00040000</td> <td>GETMAIN/FREEMAIN trace</td> <td>RADB16</td> </tr> <tr> <td>00008000</td> <td>I/O buffer trace</td> <td>RADB21</td> </tr> <tr> <td>00004000</td> <td>Write to operator (WTO) all dynamic allocation parameters</td> <td>RADB22</td> </tr> <tr> <td>00000080</td> <td>RPL trace - long</td> <td>RPLOUT</td> </tr> <tr> <td>00000040</td> <td>RPL trace - short</td> <td>RPLOUT</td> </tr> <tr> <td>00000020</td> <td>Version 2 Session Trace</td> <td>RADB33</td> </tr> <tr> <td>00000008</td> <td>Logon exit trace</td> <td>RADB35</td> </tr> <tr> <td>00000004</td> <td>Logon processor trace</td> <td>RADB36</td> </tr> <tr> <td>00000002</td> <td>SCIP exit trace</td> <td>RADB37</td> </tr> </tbody> </table>	DEBUG Setting	Trace Type	Output DLBL Name	80000000	COPY Routine and RUN TASK trace	RADB01	20000000	Display of final TCQE from DMCBSUBM	TCQE	10000000	Full TPCB/SYMBOLICS from DMCBSUB	DMCBSUB	08000000	Session manager trace.	RADB05	04000000	Separate trace per task. Example: "R0000005" to trace TASK 5	Rnnnnnn	02000000	API session trace	RADB07	01000000	DMGCSUB trace	RADB08	00400000	TCQSH from DMCBCOPY	DMCBCOP	00040000	GETMAIN/FREEMAIN trace	RADB16	00008000	I/O buffer trace	RADB21	00004000	Write to operator (WTO) all dynamic allocation parameters	RADB22	00000080	RPL trace - long	RPLOUT	00000040	RPL trace - short	RPLOUT	00000020	Version 2 Session Trace	RADB33	00000008	Logon exit trace	RADB35	00000004	Logon processor trace	RADB36	00000002	SCIP exit trace	RADB37
DEBUG Setting	Trace Type	Output DLBL Name																																																					
80000000	COPY Routine and RUN TASK trace	RADB01																																																					
20000000	Display of final TCQE from DMCBSUBM	TCQE																																																					
10000000	Full TPCB/SYMBOLICS from DMCBSUB	DMCBSUB																																																					
08000000	Session manager trace.	RADB05																																																					
04000000	Separate trace per task. Example: "R0000005" to trace TASK 5	Rnnnnnn																																																					
02000000	API session trace	RADB07																																																					
01000000	DMGCSUB trace	RADB08																																																					
00400000	TCQSH from DMCBCOPY	DMCBCOP																																																					
00040000	GETMAIN/FREEMAIN trace	RADB16																																																					
00008000	I/O buffer trace	RADB21																																																					
00004000	Write to operator (WTO) all dynamic allocation parameters	RADB22																																																					
00000080	RPL trace - long	RPLOUT																																																					
00000040	RPL trace - short	RPLOUT																																																					
00000020	Version 2 Session Trace	RADB33																																																					
00000008	Logon exit trace	RADB35																																																					
00000004	Logon processor trace	RADB36																																																					
00000002	SCIP exit trace	RADB37																																																					

Optional Parameters

There are no optional parameters for the MODIFY command.

Using MODIFY Through Batch Interface

To use the MODIFY command from the batch interface, place the desired commands in the DMBATCH job stream and submit the job while Connect:Direct is running. The following examples demonstrate the use of the MODIFY command.

The following command turns on the short RPLOUT trace:

```
MODIFY BITS.ON = X'00000040'
```

The following command turns off the short RPLOUT trace:

```
MODIFY BITS.OFF = X'00000040'
```

This example includes one of the DLBL and ASSGN cards necessary for tracing. You can use any system assignment which is not currently in use. An example can be found on the Sterling Commerce home page at <http://www.csg.stercomm.com> under Connect:Direct for VSE/ESA Frequently Asked Questions (FAQ).

```
// DLBL RAEBD01, 'SYSOUT, SYS050'
// ASSGN SYS050, SYSLST
```

Stopping Connect:Direct for VSE/ESA

The STOP CD command stops Connect:Direct operation by means of a forced, immediate, quiescent, or step-wise shutdown. A message will inform you of the pending shutdown, except when a forced shutdown is used. The STOP CD command is usually issued for system maintenance.

Command Format

The STOP CD command has the following format and associated parameters. The default value for the parameter is underlined.

Label	Command	Parameters
(optional)	STOP CD	[Force Immediate <u>Quiesce</u> Step]

Required Parameters

There are no required parameters for the STOP CD command.

Optional Parameters

The following parameters are optional for the STOP CD command.

Parameter	Description
Force	stops Connect:Direct with a user 4095 abend, and produces a sizable dump. Use this option only in a problem situation.
Immediate	specifies that all active transmissions be terminated immediately. Connect:Direct writes the statistics record, closes the files, and shuts down. All Processes resume execution when Connect:Direct is reinitialized. If a Process was checkpointed and Connect:Direct stops using the IMMEDIATE parameter, when the Process resumes execution, Connect:Direct repositions from the last checkpoint and resumes transferring data.

Parameter	Description
Quiesce	<p>specifies that all active transmissions continue to run until all steps of all executing Processes are complete. No new transmissions are started, and no additional Processes are accepted. QUIESCE is the default. All interactive sessions are terminated except for the issuer of the STOP CD command. All Processes currently running must complete, and the issuer of the command must sign off before the Connect:Direct operation stops.</p> <p>Note: When using the Connect:Direct CICS Interface with the SIGNON REENTRY set to Y, the DTF will not terminate because of an outstanding interactive task. A SHUTDOWN IMMEDIATE will terminate the DTF.</p>
Step	<p>specifies that all active transmissions run until the current Process step of each executing Process is complete. Connect:Direct then writes the statistics records, closes the files, and shuts down. All Processes resume execution when Connect:Direct is reinitialized.</p>

Using STOP CD Through Batch Interface

To use the STOP CD command from the batch interface, place your commands in the DMBATCH job stream and submit the job while Connect:Direct is running.

The following command will stop Connect:Direct, using the IMMEDIATE parameter.

```
STOP CD I
```

Using Type Key File Commands

This chapter includes discussion of the purpose of the type key file commands and presents the format and parameters of the file maintenance commands:

- ◆ INSERT TYPE Command
- ◆ UPDATE TYPE Command
- ◆ DELETE TYPE Command
- ◆ SELECT TYPE Command

Description of Type Key Defaults File

The type key defaults file contains records of file attribute information. Connect:Direct uses this information when you submit Processes with the TYPE parameter specified on the COPY statement and when Connect:Direct creates new files or accesses existing files.

The purpose of the type file is twofold.

- ◆ It saves retyping parameters such as DCB, DISP, and SPACE within Processes for files with common attributes.
- ◆ It facilitates the use of previously defined attribute specifications of different systems (for example, VM, OpenVMS, MS-DOS, and Tandem). This file is especially useful for remote Connect:Direct users who may not be familiar with VSE/ESA data set organizations and allocation parameters.

Note: The type key defaults record referenced must reside in the type defaults file on the *destination system*, which is the system responsible for allocating the new file.

Overriding File Attributes

If you specify file attributes in conjunction with the TYPE parameter on the COPY statement, the parameters coded on the COPY statement take precedence over like parameters on the associated type key defaults record. This procedure is particularly useful when you need to override a specific subparameter of a type key record that is similar to what you want.

MS-DOS and OS/2 Type Keys

There are four predefined type keys provided with Connect:Direct for VSE/ESA: TEXT, DF, DF2, and BINARY. Use these for communication with Connect:Direct for MS-DOS and OS/2 nodes. The four type keys contain file allocation information. These type keys also determine whether to perform EBCDIC to ASCII translation initiated by Connect:Direct for VSE/ESA.

Note: The TEXT type key must be installed on the VSE/ESA node in order to receive data from a Connect:Direct for MS-DOS node.

TEXT Type Key

The following is the predefined TEXT type key:

```
TYPE KEY => TEXT
          DISP = (RPL,CATLG,DELETE)      -
          DCB  = (DSORG=PS,LRECL=255,    -
                 BLKSIZE=3120,RECFM=VB) -
          SPACE = (120,(10))             -
          UNIT  = DISK
```

DF Type Key

The following is the predefined DF type key:

```
TYPE KEY => DF (Data File)
          DISP = (RPL,CATLG,DELETE)      -
          DCB  = (DSORG=PS,LRECL=255,    -
                 BLKSIZE=2554,RECFM=VB) -
          SPACE = (120,(10))             -
          UNIT  = DISK
```

DF2 Type Key

The following is the predefined DF2 type key:

```
TYPE KEY => DF2 (Data File 2)
          DISP = (RPL,CATLG,DELETE)      -
          DCB  = (DSORG=PS,LRECL=255,    -
                 BLKSIZE=3120,RECFM=VB) -
          SPACE = (120,(10))             -
          UNIT  = DISK
```


BINARY Type Key

The following is the predefined BINARY type key:

```

TYPE KEY => BINARY
DISP = (RPL,CATLG,DELETE) -
DCB = (DSORG=PS,LRECL=255, -
      BLKSIZE=6144,RECFM=U) -
SPACE = (120,(10))

```

Note: Type keys *must* be defined on all host systems with which you will be communicating with Connect:Direct for MS-DOS. Type keys are optional for all other systems.

Maintaining the Type File

The type file is maintained through the commands INSERT TYPE, UPDATE TYPE, DELETE TYPE, and SELECT TYPE. Each of these commands is presented in the following sections. These commands can be entered through batch interface or operator interface. Refer to the *Connect:Direct for VSE/ESA Operator Interface* chapter of the *Connect:Direct Console Operator's Guide* for information on using the operator interface.

Using INSERT TYPE Command

The INSERT TYPE command enables you to insert a record in the type key defaults file.

Command Format

The INSERT TYPE command uses the following parameters and subparameters. The required parameters appear in bold print.

Label	Command	Parameters
(optional)	INSert TYPE	TYPEKEY=typekey CASE=YES <u>NO</u> DCB= ([BLKSIZE = # bytes, DSORG = (PS VSAM MSAM], LRECL = # bytes, RECFM = record format)) DISP= ((<u>NEW</u> OLD RPL SHR), (,KEEP, <u>CATLG</u> , DELETE)) IOEXIT= exitname (exitname[,parameter,...]) LABEL= ([File sequence number] , [SL AL BLP LTM NL] , [password] , [IN OUT] , [RETPD=nnnn EXPDT=yyddd]) SPACE=([start track start block] , (prim)) UNIT=unit type VOL=SER=volume serial number (list) VSAMCAT=(dsn,x,x,,xxx)

Required Parameter

The following parameter is required for the Connect:Direct INSERT TYPE command.

Parameter	Description
TYPEKEY = typekey	specifies the name to be associated with the entry being added. The typekey is a 1-8-character alphanumeric string, with the first character alphabetic.

Optional Parameters

Following are the optional parameters for the INSERT TYPE command.

Parameter	Description
CASE=YES <u>NO</u>	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.

Parameter	Description
DCB = ([BLKSIZE = # bytes, DSORG = [PS VSAM MSAM] , LRECL = no.bytes, RECFM = record format])	<p>specifies DCB information to be associated with the DSN on the COPY statement.</p> <p>BLKSIZE specifies the length of the block in bytes.</p> <p>DSORG specifies the file organization. File organizations supported are PS, VSAM, and MSAM.</p> <p>LRECL specifies the length of the records in bytes.</p> <p>RECFM specifies the format of the records in the file. Any valid format, such as F, FB, U, V, and VB can be specified.</p> <p>Note: The file attributes coded on the COPY statement take precedence over the Type file. If not coded on the COPY statement, the attributes coded in the type key defaults file take precedence. If the attributes are coded neither on the COPY TO statement nor in the type key defaults file, the attributes are taken from the FROM side (source).</p> <p>Default: Source</p>
DISP = ([<u>NEW</u> OLD RPL SHR] [, KEEP , <u>CATLG</u> , DELETE])	<p>specifies the default destination file status on the receiving node. The first subparameter of DISP specifies the status of the file. Only the OLD and RPL dispositions apply to VSAM files. NEW is the default.</p> <p>NEW specifies that the Process step will create the destination file.</p> <p>OLD specifies that the destination file existed before the Process began executing and that the Process will be given exclusive control of the file. The destination file can be a VSAM file or a SAM file.</p> <p>RPL specifies for sequential files that the destination file will replace any existing file or allocate a new file.</p> <p>Note: DISP=RPL can also be specified for VSAM files, but if the file is VSAM, it must have been defined with the REUSE attribute.</p> <p>SHR specifies that the source file existed before the Process began executing and that the file can be used simultaneously by another job or Process.</p> <p>The second subparameter of DISP specifies the normal termination disposition. This subparameter does not apply to VSAM files. CATLG is the default.</p> <p>KEEP specifies that the system will keep the file after the Process step has been completed. If DISP=(NEW,KEEP), a volume serial number also must be specified.</p> <p>CATLG specifies that the system will keep the file after the Process step has been completed and that an entry will be placed in the catalog.</p> <p>DELETE specifies that the system will delete the file after the Process step has been completed.</p>

Parameter	Description
IOEXIT = exitname (exitname [, (parameter, parameter,...)])	<p>indicates that a user-written program is to be given control to perform I/O requests for the associated data. There is no default for this parameter.</p> <p>exitname specifies the name of the user-written program to be given control for I/O requests for the associated data.</p> <p>parameter (parameter, parameter,...) specifies a parameter, or list of parameters, to be passed to the specified exit. A parameter consists of a data type followed by the value in single quotes, such as C'ABC'.</p>
LABEL = ([File sequence number] , [SL AL BLP LTM NL] , [password] , [IN OUT] , [RETPD = nnnn EXPDT = yyddd])	<p>specifies information about label processing. Subparameters a positional and, if omitted, must be replaced by commas. There is no default.</p> <p>File sequence number specifies the number of the file on the tape.</p> <p>[SL AL BLP LTM NL] is the standard specification for the type of label processing.</p> <p>password specifies the password for the file, if necessary.</p> <p>[IN OUT] specifies whether a BSAM data set will be opened or whether the BDAM data set is to be read only.</p> <p>IN specifies that a BSAM data set be opened for INOUT or a BDAM data set opened for UPDAT is to be read only.</p> <p>OUT specifies that a BSAM data set opened for OUTIN or OUTINX is to be written only.</p> <p>[RETPD = nnnn EXPDT = yyddd] indicates the standard specification for retention period or expiration date of the file.</p>
SPACE = ([start track start block] , (prim))	<p>specifies the amount of DASD storage to be allocated for new files on the destination node. SPACE must be specified for all new non-VSAM files unless a TYPE that includes SPACE parameters has been specified. There is no default. Valid choices for this parameter are as follows:</p> <p>start track designates the starting track on CKD devices for the specified file.</p> <p>start block designates the starting block on FBA devices for the specified file.</p> <p>prim specifies the primary allocation of storage, either tracks or blocks.</p>
UNIT = unit type	<p>specifies the unit address, device type, or user-assigned group name that contains the data. For SAM-to-SAM copies where the destination file is new and the UNIT parameter is not coded with the TO parameter, the device type from the source file is used. The default is source.</p>
VOL = SER = volume serial number (list)	<p>specifies the volume serial number (or a list of volume serial numbers) containing the file. If VOL=SER is not specified with the FROM parameter, the file must be cataloged. There is no default.</p>

Parameter	Description
VSAMCAT = (dsn, mode, userid, pswd, cuu)	<p>specifies the VSAM catalog in which the VSAM file resides. There are no defaults</p> <p>dsn refers to the name of the catalog in which the file resides. This subparameter is required.</p> <p>mode refers to the VM file mode of the catalog. This subparameter is included for compatibility with the Connect:Direct for VM product. Any 2-character alphanumeric value is accepted.</p> <p>userid refers to the VM userid that owns the catalog. This subparameter is included for compatibility with the Connect:Direct for VM product. Any 8-character alphanumeric value is accepted.</p> <p>pswd refers to the user password of the VM user that owns the catalog. This subparameter is included for compatibility with the Connect:Direct for VM product. The value can be left blank.</p> <p>cuu refers to the device address of the VSAM catalog. This subparameter is included for compatibility with the Connect:Direct for VM product. Any valid unit number is accepted.</p> <p>Note: Only the dsn subparameter of the VSAMCAT parameter is required. Specify the following as a standard: VSAMCAT= (dsn, 1, 1, , 111)</p>

Using INSERT TYPE Through Batch Interface

To use the INSERT TYPE command from the batch interface, place your commands in a batch job stream such as DMBATCH and submit the job while Connect:Direct is running. For more information on using DMBATCH, refer to the *Utility Programs* chapter of the *Connect:Direct for VSE/ESA User's Guide*.

The following example will add a type defaults record named NEWALLOC to the type key defaults file:

```
INSERT TYPE TYPEKEY=NEWALLOC -
      DCB= (DSORG=PS) -
      DISP= (NEW, CATLG) -
      UNIT=3380
```

The NEWALLOC type record can then be referenced in a COPY request to allocate a new physical sequential file that will reside on a 3380 unit device and be cataloged upon normal termination.

Using UPDATE TYPE Command

The UPDATE TYPE command enables you to update a record in the type key defaults file. The UPDATE TYPE command uses the following format and associated parameters. The required

parameters and keywords appear in bold print. Default values for parameters and subparameters are underlined.

Command Format

The UPDATE TYPE command uses the following parameters and subparameters. The required parameters appear in bold print.

Label	Command	Parameters
(optional)	UPDate TYPE	TYPEKEY=typekey CASE=YES <u>NO</u> DCB= ([BLKSIZE = # bytes, DSORG = (PS VSAM MSAM), LRECL = # bytes, RECFM = record format]) DISP=((<u>NEW</u> OLD RPL SHR), (,KEEP, <u>CATLG</u> , DELETE)) IOEXIT=exitname (exitname[,parameter,...]) LABEL = ([File sequence number], [SL AL BLP LTM NL], [password], [IN OUT], [RETPD=nnnn EXPDT=yyddd]) SPACE=([start track start block] ,(prim)) UNIT=unit type VOL=SER=volume serial number (list)

Required Parameter

The following is the required parameter for the Connect:Direct UPDATE TYPE command.

Comand	Parameter
TYPEKEY = typekey	specifies the name to be associated with the record being updated. The typekey is a 1-8-character alphanumeric string, with the first character alphabetic.

Optional Parameters

The following parameters are the optional parameters for the Connect:Direct UPDATE TYPE command.

Parameter	Description
CASE=YES <u>NO</u>	<p>specifies whether lowercase or mixed-case data is permitted for the USER, SNODE, and FNAME subparameters. The CASE subparameter overrides the global CASE option defined at signon for the purpose of the SELECT STATISTICS command.</p> <p>YES folds the data in USER, SNODE, and FNAME to uppercase regardless of the actual data specified.</p> <p>NO preserves the actual case entered for the USER, SNODE, and FNAME subparameters.</p> <p>The case defaults to the setting defined within the session defaults if nothing is specified.</p>
DCB = ([BLKSIZE = # bytes, DSORG = [PS VSAM MSAM] , LRECL = # bytes, RECFM = record format])	<p>specifies DCB information to be associated with the DSN on the COPY statement.</p> <p>BLKSIZE specifies the length in bytes of the block.</p> <p>DSORG specifies the file organization. File organizations supported are PS, VSAM, and MSAM.</p> <p>LRECL specifies the length in bytes of the records.</p> <p>RECFM specifies the format of the records in the file. Any valid format, such as F, FB, U, V, and VB can be specified.</p> <p>Note: The file attributes coded on the COPY statement take precedence over the Type file. If not coded on the COPY statement, the attributes coded in the type key defaults file take precedence. If the attributes are coded neither on the COPY TO statement nor in the type key defaults file, the attributes are taken from the FROM side (source).</p> <p>Default: Source</p>

Parameter	Description
DISP = ([<u>NEW</u> OLD RPL SHR] [, KEEP , <u>CATLG</u> , DELETE])	<p>specifies the default destination file status on the receiving node. The first subparameter of DISP specifies the status of the file. Only the OLD and RPL dispositions apply to VSAM files. NEW is the default.</p> <p>NEW specifies that the Process step will create the destination file.</p> <p>OLD specifies that the destination file existed before the Process began executing and that the Process will be given exclusive control of the file. The destination file can be a VSAM file or a SAM file.</p> <p>RPL specifies for sequential files that the destination file will replace any existing file or allocate a new file.</p> <p>Note: DISP=RPL can also be specified for VSAM files, but if the file is VSAM, it must have been defined with the REUSE attribute.</p> <p>SHR specifies that the source file existed before the Process began executing and that the file can be used simultaneously by another job or Process.</p> <p>The second subparameter of DISP specifies the normal termination disposition. This subparameter does not apply to VSAM files. CATLG is the default.</p> <p>KEEP specifies that the system will keep the file after the Process step has been completed. If DISP=(NEW,KEEP), a volume serial number also must be specified.</p> <p>CATLG specifies that the system will keep the file after the Process step has been completed and that an entry will be placed in the catalog.</p> <p>DELETE specifies that the system will delete the file after the Process step has been completed.</p>
IOEXIT = exitname (exitname [, (parameter, parameter,...)])	<p>indicates that a user-written program is to be given control to perform I/O requests for the associated data. There is no default for this parameter.</p> <p>exitname specifies the name of the user-written program to be given control for I/O requests for the associated data.</p> <p>parameter (parameter, parameter,...) specifies a parameter, or list of parameters, to be passed to the specified exit. A parameter consists of a data type followed by the value in single quotes, such as C'ABC'.</p>

Parameter	Description
LABEL = ([File sequence number] , [SL AL BLP LTM NL] , [password] , [IN OUT] , [RETPD = nnnn EXPDT = yyddd])	<p>specifies information about label processing. Subparameters a positional and, if omitted, must be replaced by commas. There is no default.</p> <p>File sequence number specifies the number of the file on the tape.</p> <p>[SL AL BLP LTM NL] is the standard specification for the type of label processing.</p> <p>password specifies the password for the file, if necessary.</p> <p>[IN OUT] specifies whether a BSAM data set will be opened or whether the BDAM data set is to be read only.</p> <p>IN specifies that a BSAM data set be opened for INOUT or a BDAM data set opened for UPDAT is to be read only.</p> <p>OUT specifies that a BSAM data set opened for OUTIN or OUTINX is to be written only.</p> <p>[RETPD = nnnn EXPDT = yyddd] indicates the standard specification for retention period or expiration date of the file.</p>
SPACE = ([start track start block] , (prim))	<p>specifies the amount of DASD storage to be allocated for new files on the destination node. SPACE must be specified for all new non-VSAM files unless a TYPE that includes SPACE parameters has been specified. There is no default. Valid choices for this parameter are as follows:</p> <p>start track designates the starting track on CKD devices for the specified file.</p> <p>start block designates the starting block on FBA devices for the specified file.</p> <p>prim specifies the primary allocation of storage, either tracks or blocks.</p>
VOL = SER = volume serial number (list)	<p>specifies the volume serial number (or a list of volume serial numbers) containing the file. If VOL=SER is not specified with the FROM parameter, the file must be cataloged. There is no default.</p>

Using UPDATE TYPE Through Batch Interface

To use the UPDATE TYPE command from the batch interface, place your commands in a batch job stream such as DMBATCH and submit the job while Connect:Direct is running. If you have the Extended Submit Facility (ESF) available, Connect:Direct does not need to be running in order to submit the job. For more information on using DMBATCH, refer to the *Using Utility Programs* chapter of the *Connect:Direct for VSE/ESA User's Guide*.

Use the following commands to update a record in the type key defaults file. When referring to the type key defaults file record, NEWALLOC, the destination file will be an existing PS file to be allocated on a 3350 disk pack.

```
UPDATE TYPE TYPEKEY = NEWALLOC UNIT = 3350
```

Using DELETE TYPE Command

The DELETE TYPE command enables you to delete a record from the type key defaults file. The DELETE TYPE command uses the following format and associated parameters. The required parameters and keywords appear in bold print.

Command Format

The DELETE TYPE command uses the following parameters and subparameters. The required parameters appear in bold print.

Label	Command	Parameters
(optional)	DELeTe TYPE	WHERE (TYPEKEY=(typekey) (list)) CASE=YES <u>NO</u>

Required Parameter

The following parameter is required for the DELETE TYPE command.

Command	Parameter
WHERE (TYPEKEY = (typekey) (list))	<p>specifies which records(s) in the type key defaults file to delete. There is no default for this parameter.</p> <p>typekey specifies the name to be associated with the record being deleted. The typekey is a 1-8 alphanumeric character string, with the first character alphabetic.</p> <p>list specifies multiple typekeys. A list of typekeys can be specified by enclosing them in parentheses.</p>

Optional Parameters

There is one optional parameter for DELETE TYPE.

Command	Parameter
CASE=YES <u>NO</u>	<p>specifies whether lowercase or mixed-case data is permitted for the USER, SNODE, and FNAME subparameters. The CASE subparameter overrides the global CASE option defined at signon for the purpose of the SELECT STATISTICS command.</p> <p>YES folds the data in USER, SNODE, and FNAME to uppercase regardless of the actual data specified.</p> <p>NO preserves the actual case entered for the USER, SNODE, and FNAME subparameters.</p> <p>The case defaults to the setting defined within the session defaults if nothing is specified.</p>

Using DELETE TYPE Through Batch Interface

To use the DELETE TYPE command from the batch interface, place your commands in a batch job stream such as DMBATCH and submit the job while Connect:Direct is running. If you have the Extended Submit Facility (ESF) available, Connect:Direct does not need to be running in order to submit the job. For more information on using DMBATCH, refer to the *Using Utility Programs* chapter of the *Connect:Direct for VSE/ESA User's Guide*.

For example, the following commands will delete type records keyed by the names MYALLOC, NEWALLOC, and RPLALLOC from the type key defaults file.

```

/* DELETES A SINGLE TYPE RECORD */
DELETE TYPE WHERE (TYPEKEY = MYALLOC)
/* DELETES MULTIPLE TYPE RECORDS */
DELETE TYPE WHERE (TYPEKEY = (NEWALLOC RPLALLOC))

```

Using SELECT TYPE Command

Default data set attributes can be stored in the type file with a specific TYPEKEY and then referred to by a TYPEKEY name on the COPY statement. The SELECT TYPE command enables you to examine a record in the type key defaults file. The SELECT TYPE command uses the following format and associated parameters. The required parameters and keywords appear in bold print. Default values for parameters and subparameters are underlined.

Command Format

The SELECT TYPE command uses the following parameters and subparameters. The required parameters appear in bold print.

Label	Command	Parameters
(optional)	SELEct TYPE	WHERE (TYPEKEY= typekey generic (list))

Required Parameters

The following parameters are required for the SELECT TYPE command.

Parameter	Description
WHERE (TYPEKEY = typekey generic (list))	<p>specifies which records(s) in the type key defaults file to select. There is no default for this parameter.</p> <p>typekey specifies the name to be associated with the record being selected. The typekey name was created when the Entry was originally added to the Type File. The typekey is a 1-8 alphanumeric character string, with the first character alphabetic. The typekey name was created when the ENTRY was originally added to the TYPE file.</p> <p>generic specifies generic selection of type key(s). To specify type keys generically, type a one to seven alphanumeric character string, with the first character alphabetic, plus an asterisk (*). For instance, if your network includes the type keys SENDDAY, SENDMO, SENDWK, a specification of SEND* will provide information about those keys.</p> <p>list specifies multiple type keys. A list of type keys can be specified by enclosing them in parentheses and separating them by commas or spaces or both.</p>

Optional Parameters

There are no optional parameters for SELECT TYPE.

Using SELECT TYPE Through Batch Interface

To use the SELECT TYPE command from the batch interface, place your commands in a batch job stream like the example presented in the *Utility Programs* chapter of the *Connect:Direct for VSE/ESA User's Guide*, and submit the job while Connect:Direct is running. If you have the Extended Submit Facility (ESF) available, Connect:Direct does not need to be running in order to submit the job.

```
SELREC SELECT TYPE WHERE (TYPEKEY=ALLOCATE)
```

Maintaining User Authorization File

This chapter includes the following:

- ◆ Description of the authorization facility
- ◆ Maintenance of the user authorization file

Read Chapter 6, *Planning and Controlling Security* for more information about the authorization facility.

Authorization Facility

The Connect:Direct authorization facility controls access to Connect:Direct functions and serves as the source of security information. If you use the Connect:Direct authorization facility, you must identify all Connect:Direct users in all nodes that will participate in Process execution.

Authorization File

The authorization file contains records of user attribute defaults. Each of these records defines which features of Connect:Direct the user is allowed to access for each node.

Maintaining the User Authorization File

The user authorization file maintenance commands can be entered through:

- ◆ Batch Interface
- ◆ Operator Interface

Refer to the *Connect:Direct for VSE/ESA Operator Interface* chapter of the *Connect:Direct Console Operator's Guide* for information on using the Operator Interface.

There are four user authorization file maintenance commands:

- ◆ INSERT USER inserts a user record in the authorization file.
- ◆ UPDATE USER updates a user record in the authorization file.
- ◆ DELETE USER deletes a user record from the authorization file.

- ◆ SELECT USER selects a user record from the authorization file.

When you receive authorization to use the Connect:Direct software, you also receive authorization to use its functions. Currently, the following functions are available for authorization purposes:

- ◆ CHANGE PROCESS
- ◆ SELECT PROCESS
- ◆ SELECT NETWORK MAP
- ◆ DELETE TYPE
- ◆ SELECT PROCESS
- ◆ DELETE USER
- ◆ SELECT STATISTICS
- ◆ FLUSH PROCESS
- ◆ SELECT TASK
- ◆ FLUSH TASK
- ◆ SELECT TYPE
- ◆ INSERT/UPDATE TYPE
- ◆ SELECT USER
- ◆ INSERT/UPDATE USER
- ◆ SUBMIT PROCESS
- ◆ MODIFY (TRACE)
- ◆ SUSPEND PROCESS
- ◆ STOP CD

Using INSERT USER Command

The INSERT USER command enables you to add a user to the Connect:Direct Authorization file. The Connect:Direct functions and resources available to the user are defined through this command.

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

Command Format

The INSERT USER command uses the parameters listed in the following figure. Required parameters are in bold type, and defaults are underlined. The following pages define each parameter.

Label	Command	Parameters
(optional)	INSert USER	NAME='user name'
		USERID=(nodeid, userid)
		ADD TYPE=Y <u>N</u>
		ALTER TYPE=Y <u>N</u>
		READ TYPE=Y <u>N</u>
		REMOVE TYPE=Y <u>N</u>
		ADD USER=Y <u>N</u>
		ALTER USER=Y <u>N</u>
		READ USER=Y <u>N</u>
		REMOVE USER=Y <u>N</u>
		CASE=YES <u>NO</u>
		CHANGE=Y <u>N</u>
		COPY=Y <u>N</u>
		DELPR=Y <u>N</u>
		FLUSH=Y <u>N</u>
		MAXSA=max signon attempts
		MODALS=Y <u>N</u>
		MODIFY=Y <u>N</u>
		PASSword=initial password
		PHone='phone number'
		RUNJOB=Y <u>N</u>
		RUNTASK=Y <u>N</u>
		SECURITY=(security id, security pswd)
		SELNET=Y <u>N</u>
		SELPR=Y <u>N</u>
		SELSTAT=Y <u>N</u>
		STOPCD=Y <u>N</u>
		SUBMIT =Y <u>N</u>
		SUBMITTER.CMDS=(Y <u>N</u> Y <u>N</u> Y <u>N</u> Y <u>N</u> Y <u>N</u>)

Required Parameters

Command	Parameter
NAME = 'username'	specifies the user's full name. The NAME is a string of 1-20 alphanumeric characters. If blanks are embedded in the NAME parameter, the NAME must be enclosed in single quotation marks.
USERID = (nodeid, userid)	specifies the user node and userid of the record being added. The user node is a one to sixteen alphanumeric character string. The userid can contain one to eight characters of any kind. nodeid specifies the user node of the user record. userid specifies the userid of the user record.

Optional Parameters

The following optional parameters for the INSERT USER command are separated into two categories:

- ◆ Authorization record parameters
- ◆ Functional authorization parameters

Authorization Record Parameters

The following are the authorization record parameters for the INSERT USER command. A user can be authorized to add, alter, read, or remove a record. The authorization is specified by indicating the action (ADD, ALTER, READ, REMOVE), followed by the record type. If no action is specified for a type or user record, the default value is No.

Command	Parameter
ADD TYPE = Y <u>N</u>	specifies whether the user is allowed to insert new records into the Type Defaults file. The default value is No.
ALTER TYPE = Y <u>N</u>	specifies whether the user is allowed to update records in the Type Defaults file. The default value is No.
READ TYPE = Y <u>N</u>	specifies whether the user is allowed to read records from the Type Defaults file. The default value is No.
REMOVE TYPE = Y <u>N</u>	specifies whether the user is allowed to delete records from the Type Defaults file. The default value is No.
ADD USER = Y <u>N</u>	specifies whether the user is allowed to insert new records into the Authorization file. The default value is No.
ALTER USER = Y <u>N</u>	specifies whether the user is allowed to update records in the Authorization file. The default value is No.
READ USER = Y <u>N</u>	specifies whether the user is allowed to read records from the Authorization file. The default value is No.

Command	Parameter
REMOVE USER = Y <u>N</u>	specifies whether the user is allowed to delete records from the Authorization file. The default value is No.

Functional Authorization Parameters

The following are the functional authorization parameters for the INSERT USER command:

Command	Parameter
CASE=YES <u>NO</u>	<p>specifies whether lowercase or mixed-case data is permitted for the USER, SNODE, and FNAME subparameters. The CASE subparameter overrides the global CASE option defined at signon for the purpose of the SELECT STATISTICS command.</p> <p>YES folds the data in USER, SNODE, and FNAME to uppercase regardless of the actual data specified.</p> <p>NO preserves the actual case entered for the USER, SNODE, and FNAME subparameters.</p> <p>The CASE defaults to the setting defined within the session defaults if nothing is specified.</p>
CHANGE = Y <u>N</u>	specifies whether the user is allowed to use the CHANGE PROCESS command. The default value is No.
COPY = Y <u>N</u>	specifies whether the user is allowed to use the COPY statement. The default value is No.
DELPR = Y <u>N</u>	specifies whether the user is allowed to use the DELETE PROCESS command. The default value is No.
FLUSH = Y <u>N</u>	specifies whether the user is allowed to use the FLUSH PROCESS and SUSPEND PROCESS commands. The default value is No.
MAXSA = max signon attempts	specifies the maximum number of signon attempts the user is allowed per hour. The range is from 0-99. The default is 60. There is no maximum number if 0 is specified.
MODALS = Y <u>N</u>	specifies whether the user is allowed to use the modal statements IF THEN, ELSE, EIF, GOTO, and EXIT. The default value is No.
MODIFY = Y <u>N</u>	specifies whether the user is allowed to request traces and modify certain Connect:Direct operational functions. The default is No.
PASSword = initial password	<p>defines the initial password for the userid. The password is a 4-8-character alphanumeric string.</p> <p>Note: IBM only recognizes uppercase and Tandem users cannot use the escape or control keys because the Connect:Direct software does not recognize them.</p>
PHone = 'phone number'	specifies the phone number of the user. The phone number must be enclosed in single quotation marks. The quotation marks allow for a space after the area code. The length can be from 1-14 characters.

Command	Parameter
RUNJOB = Y <u>N</u>	specifies whether the user is allowed to use the RUN JOB statement. The default value is No.
RUNTASK = Y <u>N</u>	specifies whether the user is allowed to use the RUN TASK statement. The default value is No.
SECURITY = (security id, security pswd)	<p>specifies the security ID and security password used to identify the user's file authorization.</p> <p>security id specifies the one to eight character security system ID for the user. This ID must meet the standards of the security subsystem at the user's site. The security ID is required if this parameter is specified.</p> <p>security pswd specifies the one to eight character security system password for the user. This password must meet the standards of the security subsystem at the user's site.</p>
SELNET = Y <u>N</u>	specifies whether the user is allowed to use the SELECT NETMAP command. The default value is No.
SELPR = Y <u>N</u>	specifies whether the user is allowed to use the SELECT PROCESS command. The default value is No.
SELSTAT = Y <u>N</u>	specifies whether the user is allowed to use the SELECT STATISTICS command. The default value is No.
STOPCD = Y <u>N</u>	specifies whether the user is allowed to use the STOP Connect:Direct command. The default value is No.
SUBMIT = Y <u>N</u>	specifies whether the user is allowed to use the SUBMIT statement to define and submit processes for execution. The default value is No.
SUBMITTER.CMDS = (Y <u>N</u> Y <u>N</u> Y <u>N</u> Y <u>N</u>)	<p>specifies whether the user is allowed to issue certain commands concerning the Processes that the user submitted. These commands, in order, are as follows: SELECT PROCESS, DELETE PROCESS, FLUSH (or SUSPEND) PROCESS, CHANGE PROCESS, and SELECT STATISTICS. The default value is No.</p> <p>Note: The Connect:Direct CICS interface does not reference the SUBMITTER.CMDS field.</p>

Using INSERT USER Through Batch Interface

To use the INSERT USER command from the batch interface, place your commands in a batch job stream such as DMBATCH and submit the job while the Connect:Direct software is running. If you have the extended submit facility (ESF) available, the Connect:Direct system does not need to be running in order to submit the job. For more information on using DMBATCH, refer to the *Using Utility Programs* chapter of the *Connect:Direct for VSE/ESA User's Guide*.

The following example shows how to add a user record for userid SMITH to the authorization file:

```

SIGNON          USERID=(userid, password)
INSERT USER          USERID = (DALLAS, SMITH)          -
                   NAME = 'RB SMITH'  PASS = XYZZY    -
                   PH = '214 555-5555'              -
                   ADD USER = Y  ALTER USER = Y      -
                   READ USER = Y  REMOVE USER = Y   -
                   SUBMIT = Y                      -
                   SUBMITTER.CMDS = (Y Y Y Y N)
SIGNOFF

```

With the above definition, Smith can perform the following functions:

- ◆ Add users to the Connect:Direct authorization file.
- ◆ Update and read user records.
- ◆ Delete user records.
- ◆ Define and submit processes for execution.
- ◆ Select, delete, flush/suspend, and change processes that have been submitted.

Smith cannot perform the SELECT STATISTICS command on any processes, however, regardless of who submitted them. Smith's initial password is XYZZY and Smith's phone number is (214) 555-5555.

Using UPDATE USER Command

The UPDATE USER command enables you to update a user record in the Connect:Direct authorization file. The Connect:Direct functions and resources available to the user are defined through this command.

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

Command Format

The UPDATE USER command uses the parameters listed in the following figure. Required parameters are in bold type, and defaults are underlined. The following pages define each parameter.

Label	Command	Parameters
(optional)	UPDate USER	USERID =(<u>nodeid</u> , <u>userid</u>) ADD TYPE=Y <u>N</u> ALTER TYPE=Y <u>N</u>

Label	Command	Parameters
	UPDate USER (continued)	READ TYPE=Y <u>N</u>
		REMOVE TYPE=Y <u>N</u>
		ADD USER=Y <u>N</u>
		ALTER USER=Y <u>N</u>
		READ USER=Y <u>N</u>
		REMOVE USER=Y <u>N</u>
		CASE=YES <u>NO</u>
		CHANGE=Y <u>N</u>
		COPY=Y <u>N</u>
		DELPR=Y <u>N</u>
		FLUSH=Y <u>N</u>
		MAXSA=max signon attempts
		MODALS=Y <u>N</u>
		MODIFY=Y <u>N</u>
		NAME='user name'
		PASSword=initial password
		PHone='phone number'
		RESETSA
		RUNJOB=Y <u>N</u>
		RUNTASK=Y <u>N</u>
		SECURITY=(security id, security pswd)
		SELNET=Y <u>N</u>
		SELPR=Y <u>N</u>
		SELSTAT=Y <u>N</u>
		STOPCD=Y <u>N</u>
		SUBMIT=Y <u>N</u>
		SUBMITTER.CMDS=(Y <u>N</u> Y <u>N</u> Y <u>N</u> Y <u>N</u> Y <u>N</u>)

Required Parameters

Command	Parameter
USERID = (nodeid, userid)	<p>specifies the user node and userid of the record being updated. The user node is a 1-16-character alphanumeric string. The userid can contain 1-8 characters of any kind.</p> <p>nodeid specifies the user node of the user record.</p> <p>userid specifies the userid of the user record.</p>

Optional Parameters

The following optional parameters for the UPDATE USER command are separated into two categories:

- ◆ Authorization record parameters
- ◆ Functional authorization parameters

Authorization Record Parameters

The following are the authorization record parameters for the UPDATE USER command. A user can be authorized to add, alter, read, or remove a record. The authorization is specified by indicating the action (ADD, ALTER, READ, REMOVE), followed by the record type. If no action is specified for a type or user record, the default value is No.

Command	Parameter
ADD TYPE = Y <u>N</u>	specifies whether the user is allowed to insert new records into the Type Defaults file. The default value is No.
ALTER TYPE = Y <u>N</u>	specifies whether the user is allowed to update records in the Type Defaults file. The default value is No.
READ TYPE = Y <u>N</u>	specifies whether the user is allowed to read records from the Type Defaults file. The default value is No.
REMOVE TYPE = Y <u>N</u>	specifies whether the user is allowed to delete records from the Type Defaults file. The default value is No.
ADD USER = Y <u>N</u>	specifies whether the user is allowed to insert new records into the Authorization file. The default value is No.
ALTER USER = Y <u>N</u>	specifies whether the user is allowed to update records in the Authorization file. The default value is No.
READ USER = Y <u>N</u>	specifies whether the user is allowed to read records from the Authorization file. The default value is No.
REMOVE USER = Y <u>N</u>	specifies whether the user is allowed to delete records from the Authorization file. The default value is No.

Functional Authorization Parameters

The following are the functional authorization parameters for the UPDATE USER command.

Command	Parameter
CASE=YES <u>N</u> O	<p>specifies whether lowercase or mixed-case data is permitted for the USER, SNODE, and FNAME subparameters. The CASE subparameter overrides the global CASE option defined at signon for the purpose of the SELECT STATISTICS command.</p> <p>YES folds the data in USER, SNODE, and FNAME to uppercase regardless of the actual data specified.</p> <p>NO preserves the actual case entered for the USER, SNODE, and FNAME subparameters.</p> <p>The CASE defaults to the setting defined within the session defaults if nothing is specified.</p>
CHANGE = Y <u>N</u>	<p>specifies whether the user is allowed to use the CHANGE PROCESS command. The default value is No.</p>
COPY = Y <u>N</u>	<p>specifies whether the USER is allowed to use the COPY statement. The default value is No.</p>
DELPR = Y <u>N</u>	<p>specifies whether the user is allowed to use the DELETE PROCESS command. The default value is No.</p>
FLUSH = Y <u>N</u>	<p>specifies whether the user is allowed to use the FLUSH PROCESS and SUSPEND PROCESS commands. The default value is No.</p>
MAXSA = max signon attempts	<p>specifies the maximum number of signon attempts the user is allowed per hour. The range is from 0-99. The default is 60. There is no maximum number if 0 is specified.</p>
MODALS = Y <u>N</u>	<p>specifies whether the user is allowed to use the modal statements IF THEN, ELSE, EIF, GOTO, and EXIT. The default value is No.</p>
MODIFY = Y <u>N</u>	<p>specifies whether the user is allowed to request traces and modify certain Connect:Direct operational functions. The default is No.</p>
NAME = 'username'	<p>changes the user's full name. NAME is a string of one to 20 alphanumeric characters. If blanks are embedded in the parameter, it must be enclosed in single quotation marks.</p>
PASSword = initial password	<p>defines the initial password for the userid. The password is a 4-8-character alphanumeric string.</p> <p>Note: IBM only recognizes uppercase and Tandem users cannot use the escape or control keys because Connect:Direct does not recognize them.</p>
PHone = 'phone number'	<p>specifies the phone number of the user. The phone number must be enclosed in single quotation marks. The quotation marks allow for a space after the area code.</p>
RESETSA	<p>specifies that the signon attempt count is to be reset to zero. This allows the user to try to signon, even if the user has previously exceeded the maximum number of signon attempts.</p>

Command	Parameter
RUNJOB = Y <u>N</u>	specifies whether the user is allowed to use the RUN JOB statement. The default value is No.
RUNTASK = Y <u>N</u>	specifies whether the user is allowed to use the RUN TASK statement. The default value is No.
SECURITY = (security ID, security pswd)	specifies the security ID and security password used to identify the user's file authorization. security ID specifies the 1-20-character security system ID for the user. This ID must meet the standards of the security subsystem at the user's site. The security ID is required if this parameter is specified. security pswd specifies the 1-20-character security system password for the user. This password must meet the standards of the security subsystem at the user's site.
SELNET = Y <u>N</u>	specifies whether the user is allowed to use the SELECT NETMAP command. The default value is No.
SELPR = Y <u>N</u>	specifies whether the user is allowed to use the SELECT PROCESS command. The default value is No.
SELSTAT = Y <u>N</u>	specifies whether the user is allowed to use the SELECT STATISTICS command. The default value is No.
STOPCD = Y <u>N</u>	specifies whether the user is allowed to use the STOP Connect:Direct command. The default value is No.
SUBMIT = Y <u>N</u>	specifies whether the user is allowed to use the SUBMIT statement to define and submit processes for execution. The default value is No.
SUBMITTER.CMDS = (Y <u>N</u> Y <u>N</u> Y <u>N</u> Y <u>N</u>)	specifies whether the user is allowed to issue certain commands concerning the processes that the user submitted. These commands, in order, are as follows: SELECT PROCESS, DELETE PROCESS, FLUSH (or SUSPEND) PROCESS, CHANGE PROCESS, and SELECT STATISTICS. The default value is No.

Using UPDATE USER Through Batch Interface

To use the UPDATE USER command from the batch interface, place your commands in a batch job stream such as DMBATCH and submit the job while the Connect:Direct software is running. If you have the extended submit facility (ESF) available, the Connect:Direct system does not need to be running in order to submit the job. For more information on using DMBATCH, refer to the *Using Utility Programs* chapter of the *Connect:Direct for VSE/ESA User's Guide*.

The following commands will update the record of a user named Smith in the Authorization file.

```

UPDATE USER USERID = (DALLAS, SMITH)      -
      NAME = 'RB SMITH'                    -
      PASS = XYZZY                          -
      PH = '214 555-5555'                  -
      ADD USER = Y  ALTER USER = Y         -
      READ USER = Y  REMOVE USER = Y      -
      CH = Y  FLUSH = Y  DELPR = Y

```

With the previous updates, Smith will be able to perform the following functions:

- ◆ Add users to the Connect:Direct authorization file
- ◆ Update and read user records
- ◆ Delete user records
- ◆ Change a process in the TCQ
- ◆ Delete an executing process from the TCQ
- ◆ Delete an inactive process from the TCQ

Using DELETE USER Command

The DELETE USER command removes a user record from the Connect:Direct authorization file.

Command Format

The DELETE USER command has the following format and associated parameters. The required parameters and keywords appear in bold print.

Label	Command	Parameters
(optional)	DELeTe USER	WHERE=(USERID=(nodeid, userid) (list))
		CASE=YES NO

Required Parameter

The required parameter for the DELETE USER command follows:

Command	Parameters
WHERE USERID = (nodeid, userid)	<p>specifies which user record(s) to delete.</p> <p>USERID = (nodeid, userid) specifies the record to be deleted from the Connect:Direct authorization file.</p> <p>nodeid specifies the node ID of the user record that will be searched. The nodeid is a 1- to 16-character alphanumeric string.</p> <p>userid specifies the userid of the user record. The complete userid consists of the nodeid and the userid enclosed in parentheses and separated by a comma.</p> <p>(list) specifies a list of userids.</p>

Optional Parameters

There is one optional parameter for DELETE USER.

Command	Parameters
CASE=YES <u>NO</u>	<p>specifies whether lowercase or mixed-case data is permitted for the USER, SNODE, and FNAME subparameters. The CASE subparameter overrides the global CASE option defined at signon for the purpose of the SELECT STATISTICS command.</p> <p>YES folds the data in USER, SNODE, and FNAME to uppercase regardless of the actual data specified.</p> <p>NO preserves the actual case entered for the USER, SNODE, and FNAME subparameters.</p> <p>The case defaults to the setting defined within the session defaults if nothing is specified.</p>

Using DELETE USER Through Batch Interface

To use the DELETE USER command from the batch interface, place your commands in a batch job stream such as DMBATCH and submit the job while the Connect:Direct software is running. For more information on using DMBATCH, refer to the *Using Utility Programs* chapter of the *Connect:Direct for VSE/ESA User's Guide*.

The following example shows how to delete user records.

```

/* DELETES A SINGLE User record */
DELETE USER WHERE (USERID = (MPLS, SMITH))
/* DELETES MULTIPLE User records */
DELETE USER WHERE (USERID=((DALLAS, JONES), -
(MPLS, SMITH), (CHICAGO, BROWN)))

```

Using SELECT USER Command

The SELECT USER command displays a user record in the authorization file. You can specify the search criteria and the form in which the information will be presented (such as PPrint, TABLE).

Command Format

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

Label	Command	Parameters
(optional)	SElect USER	WHERE (USERID=(nodeid, userid) generic (list)) CASE= <u>YES NO</u>

Required Parameters

The following parameter is the only required parameter for SELECT USER.

Command	Parameters
WHERE (USERID = (nodeid, userid) generic (list))	<p>specifies which User record(s) the user wants to examine.</p> <p>USERID = (nodeid, userid) generic (list) specifies the record to be searched for in the Connect:Direct Authorization file.</p> <p>nodeid specifies the node ID of the user record that will be searched. The nodeid is a 1- to 16-character alphanumeric string. If the user node is not specified, the node ID defaults to the Connect:Direct system that receives the command.</p> <p>userid specifies the userid of the user record. The complete userid consists of the nodeid and the userid enclosed in parentheses and separated by a comma.</p> <p>generic specifies generic selection of userids. To specify user nodes and userids generically, type a 1- to 7-character alphanumeric string with the first character alphabetic, plus an asterisk (*). For instance, if a userid of B* was specified, the user might examine User records for BLACK, BRADFORD, and BROWN.</p> <p>list indicates that you can specify a list of nodenames by enclosing them in parentheses and separating each by a comma or a blank.</p>

Optional Parameters

There is one optional parameter for SELECT USER.

Command	Parameters
CASE=YES <u>NO</u>	<p>specifies whether lowercase or mixed-case data is permitted for the USER, SNODE, and FNAME subparameters. The CASE subparameter overrides the global CASE option defined at signon for the purpose of the SELECT STATISTICS command.</p> <p>YES folds the data in USER, SNODE, and FNAME to uppercase regardless of the actual data specified.</p> <p>NO preserves the actual case entered for the USER, SNODE, and FNAME subparameters.</p> <p>The case defaults to the setting defined within the session defaults if nothing is specified.</p>

Using SELECT USER Through Batch Interface

To use the SELECT USER command from the batch interface, place your commands in a batch job stream such as DMBATCH and submit the job while the Connect:Direct system is running. If you have the extended submit facility (ESF) available, the Connect:Direct system does not need to be running in order to submit the job. For more information on using DMBATCH, refer to the *Using Utility Programs* chapter of the *Connect:Direct for VSE/ESA User's Guide*.

The following command searches for User DALLAS.BILL at the local (default) node.

```
SELECT USER WHERE (USERID = (, DALLAS.BILL))
```

Defining and Maintaining the Network Map

The network map identifies the local Connect:Direct node and the nodes with which it can communicate. The network map has a local node entry and one or more adjacent node entries. Each entry identifies the communications name and protocol associated with a Connect:Direct node. These logical node names are then referenced on Connect:Direct Processes.

The source form of the network map is generated during the install process and is input to the network map load utility, DMCNTMPL. This utility creates the VSAM form of the network map that Connect:Direct uses.

You must customize the network map according to environmental requirements. You must stop Connect:Direct, reload the network map, and then restart Connect:Direct before changes to the network map can take place.

To allocate and load the network map, retrieve member LDNETMAP.PROC from NDMLIB. Please read the comments and edit the member appropriately to reflect your configuration. Then, update POWER JECL and submit the job to execute.

See the *Connect:Direct for VSE/ESA Installation Guide* for rules governing the network map for cross-domain VTAM definitions.

Definitions

The following terms are used when working with your network map.

Local Node Record

The local node record defines the name of your local Connect:Direct system and the APPLID used for VTAM communication. Node-to-node sessions will be established with this APPLID (labelled A in the figure on page 64). The local node record also defines the SUPERUSR password, and the TCQ and TCX DSNs.

Adjacent Node Record

The adjacent node record represents the Connect:Direct logical node name and its associated VTAM APPLID for other Connect:Direct nodes in the network, as well as other information about that node. There is one ADJACENT.NODE entry for each node in the network that Connect:Direct can communicate with.

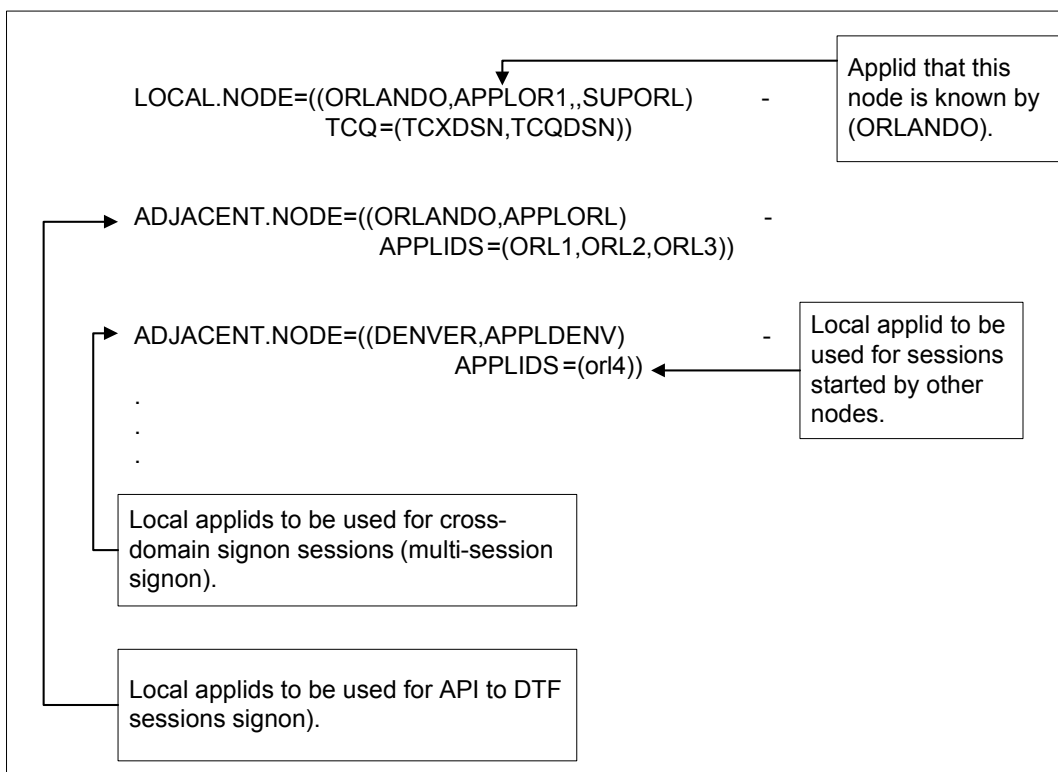
Defining the Local Node as an Adjacent Node

The local node must also be defined as an adjacent node because VTAM sessions are used to logon from the API to the DTF. The APPLIDs defined in the adjacent node record APPLIDS= keyword are used for these sessions. The number of concurrent users is limited by the number of APPLIDs defined in the adjacent node record of the local node (labeled B in the diagram on page 64).

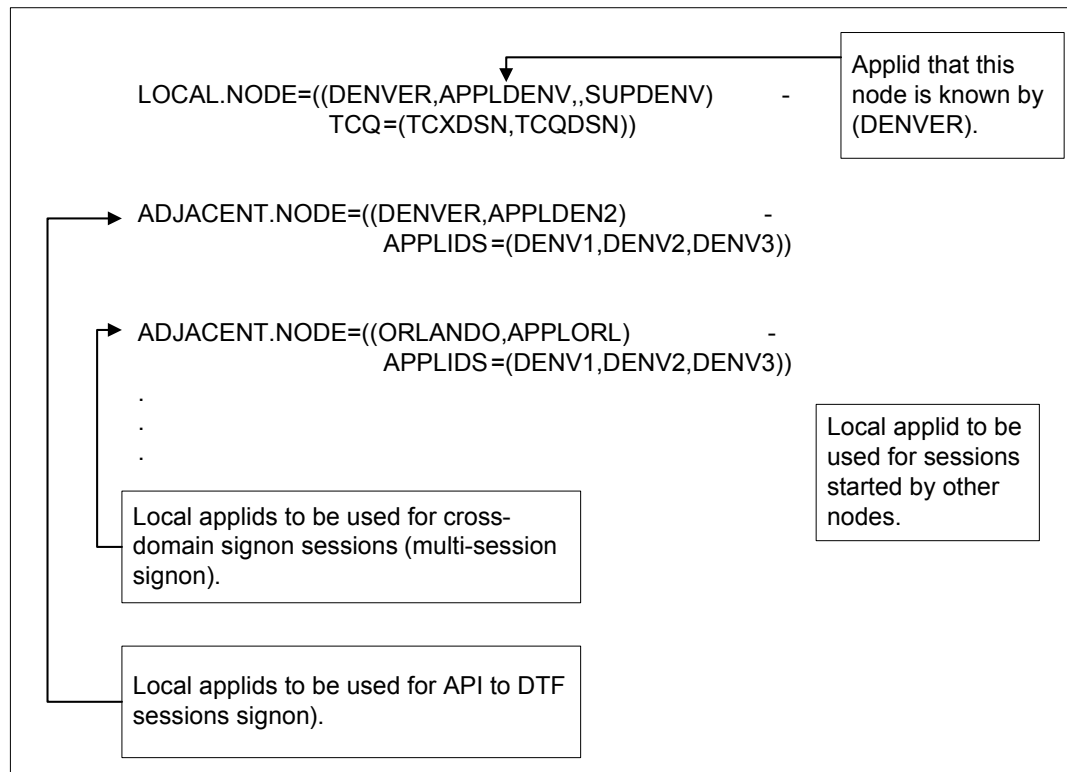
Defining the local node as an adjacent node gives you the ability to initiate sessions with your own local node, where you are both the Primary node (PNODE) and the Secondary node (SNODE).

Examples

The following figures are examples of the connection between two nodes, in relation to the network map. A portion of each network map file is shown, including the LOCAL.NODE and pertinent ADJACENT.NODE records.



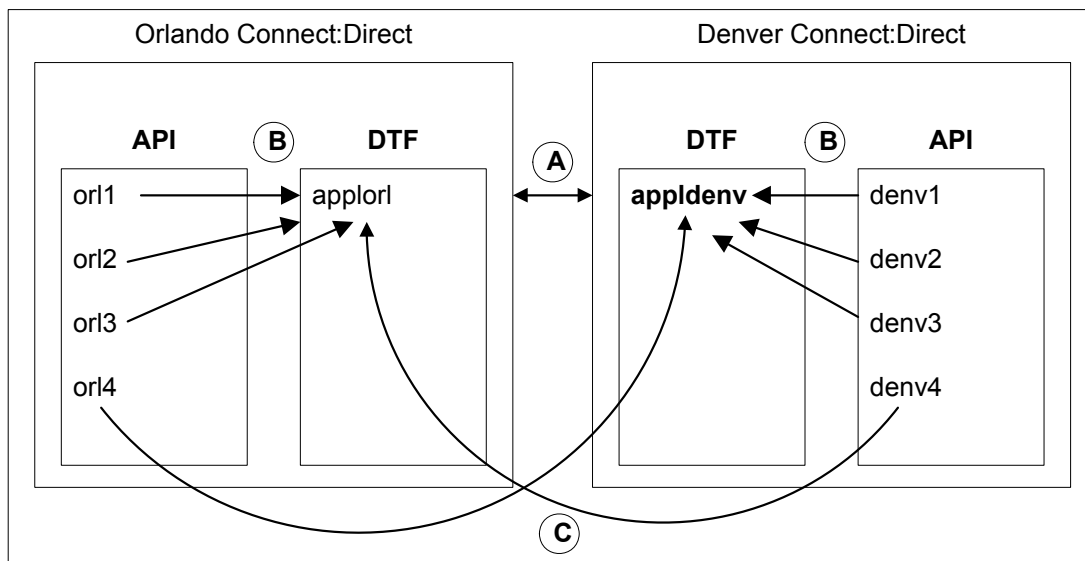
The previous figure is an example of the network map definitions for the node in Orlando. The following figure is an example of the network map definitions for the node in Denver.



A cross-domain SIGNON, a multi-session signon through Connect:Direct, labeled C in the following diagram, shows an API signon from Orlando to Denver.

The adjacent node records for other nodes identify those nodes and their applids to the local node. The adjacent node record also associates a user defined name with a VTAM APPLID used to communicate with the other node.

The following figure is a diagram of APPLID to APPLID VTAM sessions for the previous network map Configuration.



Keywords

The following two sections explain the network map keywords: LOCAL.NODE and ADJACENT.NODE.

LOCAL.NODE Keyword

The LOCAL.NODE entry defines the logical Connect:Direct name of the local Connect:Direct DTF and its communications name. The local node entry also contains the name of the transmission queue and the SUPERUSR ID password, if specified. The following paragraphs define the syntax in this figure.

```
LOCAL.NODE=( (node name,communications name,,superuser password) -
              TCQ=(tcxdsn, tcqdsn) -
              )
```

Positional Parameters

The following positional parameters define the local node entry:

Parameter	Description
node name	specifies the logical Connect:Direct name of the local node. This first positional parameter can be 1-16 alphanumeric characters in length. The local node must exist as an adjacent node entry also. For further explanation, see the sample network map entries on page 71.

Parameter	Description
communications name	is the second positional parameter. It specifies the name that Connect:Direct uses to communicate over the network. The name must be 1-8 characters long. For SNA this field must contain the VTAM APPLID that the local DTF uses for DTF-DTF communications. Note: Two commas must follow the VTAM APPLID in the LOCAL.NODE definition.
null	is the third positional parameter and is not used.
superuser password	specifies the 1-8 character SUPERUSR ID password. The initial value for this fourth positional parameter is specified during installation. When the administrator signs on to the system with the SUPERUSR ID as the system is brought up for the first time, the administrator can assign user IDs and authorizations. The superuser password provides authorization to perform all Connect:Direct commands. The SUPERUSR ID bypasses your usual security system at signon. This bypass may be necessary if Connect:Direct is configured improperly, and you cannot sign on. SUPERUSR still goes through usual data-set verification done by the Stage 2 security exit.

Keyword Parameter

The following is the keyword parameter for the network map local node entry.

Parameter	Description
TCQ=(tcxdsn,tcqdsn)	specifies the names of the two files that together make up the TCQ. tcxdsn , the first subparameter, identifies the data-set name of the TCQ index (TCX). tcqdsn , the second subparameter, identifies the data-set name of the TCQ.

Defining Local Node as Adjacent Node

The local node must also be defined as an adjacent node for the following reasons:

- ◆ To specify the VTAM application IDs to be used for IUI and batch sessions. The APPLIDS match those defined during installation preparation as described in the *Connect:Direct for VSE/ESA Installation Guide*.
- ◆ To provide the ability to run Processes where the local node is both the initiating and target node (PNODE and SNODE). The communications name matches the APPLID defined during installation preparation as described in the *Connect:Direct for VSE/ESA Installation Guide*. Observe the rules as described in *PNODE=SNODE Processing* on page 72.

ADJACENT.NODE Keyword

The ADJACENT.NODE keyword defines every remote node in the network with which Connect:Direct can communicate, including the local node. Each entry specifies a locally used Connect:Direct name, its associated network communications name, and session control parameters for these nodes.

The local node is also defined with the ADJACENT.NODE keyword to allow execution of Processes in which PNODE is the same as SNODE.

The following paragraphs define the syntax shown in the following figure.

```

ADJACENT.NODE=(
    (nodename,
      communications name,
      remote library name | IP address or Alias,
      session type,
      security node type,
      data direction restriction )
  PARSESS=(max default)
  ENVIRONMENT=operating environment
  LOGMODE=logmode entry name
  APPLIDS=(vtam applid1 [,vtam applid2,...] )
  NETID=networkid
  PNODE.LUS=(luname1 [,luname2,...] )
  SNODE.LUS=(luname1 [,luname2,...] )
  NDMPACE = (#sends,time)
)

```

Positional Parameters

The following positional parameters are for the network map adjacent node entry.

Parameter	Description
ADJACENT.NODE= (nodename, communications name, (remote library name IP address or Alias), session.type, security node type, data direction restriction)	<p>These six positional parameters further identify the node and its characteristics. Commas must be coded if a parameter is not used.</p> <p>nodename specifies the logical node name of another node with which Connect:Direct will communicate. The name can be 1-16 characters long. This first positional parameter is required.</p> <p>communications name specifies the network name of the partner Connect:Direct. It may be an SNA VTAM APPLID or a TCP/IP port number. This name must be identical to the APPLID parameter on the LOCAL.NODE keyword in the network map on that node. The name must be 1-8 characters long. This second positional parameter is optional.</p> <p>For SNA and OS/400 (independent LUs), this field must contain the VTAM APPLID the local DTF uses for DTF-DTF communications with the partner Connect:Direct.</p> <p>The communications name is the VTAM APPLID of the remote Connect:Direct node (the same name as defined for communications name in network map of the remote Connect:Direct node).</p> <p>For SNA, this APPLID should be defined as a CDRSC locally and as an APPL to VTAM on the remote system.</p> <p>For MS-DOS and OS/2, if the partner system is a remote PU2 node, the communications name is the LU name used to contact the remote Connect:Direct. This parameter is used for MS-DOS and can also be used for OS/2 LU0 or LU6.2.</p> <p>For OpenVMS, Tandem, (non NETEX), and OS/400 (LU and dependent LUs) use the PNODE.LUS and SNODE.LUS parameters and leave this field blank, because LU pooling is used.</p>

Parameter	Description										
ADJACENT.NODE (continued)	<p>For NetWare, if you are using LU6.2 independent LUs (parallel-session-capable LUs), you can specify a single LU name. In this case, only the named LU is used to communicate between the two Connect:Direct nodes, and any session limits placed on that LU impact the number of parallel sessions that can be active between the two nodes. Alternatively, you can leave this field blank and specify SNODE.LUS and PNODE.LUS.</p> <p>For OS/400 (LU and dependent LUs), OpenVMS, and Tandem the APPLID should be a null parameter because LU pooling is used. The SNODE.LUS parameter specifies a list of LUs.</p> <p>For TCP/IP, this field contains the TCP/IP port number of the remote partner Connect:Direct. The field need not be used if the partner Connect:Direct is initialized using the default TCP/IP port number. This port number refers only to the partner Connect:Direct and does not change the port number for the host Connect:Direct as defined at initialization. See the <i>Connect:Direct for VSE/ESA Installation Guide</i> for more information about TCP/IP port number.</p> <p>remote library name IP address or Alias specifies the host or library name. Only the library name is required, and this parameter is used for OS/2, OS/400, and TCP/IP nodes only.</p> <p>For OS/400 nodes, this parameter specifies the name of the library where the Connect:Direct for OS/400 program SMMAIN resides for the remote Connect:Direct.</p> <p>For TCP/IP nodes, this parameter specifies the IP address or Alias name to be used to establish a TCP/IP session.</p> <p>For OS/2 (LU6.2 only) nodes this parameter must be CDOS2.</p> <p>session type specifies the type of session communications protocol to use for communications with this adjacent node. This fourth positional parameter is required for OS/400 adjacent nodes and any node using a protocol other than LU0. Valid values are described in the following table.</p> <table border="1"> <thead> <tr> <th>Session Type</th> <th>Session Protocol</th> </tr> </thead> <tbody> <tr> <td>SNA</td> <td>LU0</td> </tr> <tr> <td>SNUF</td> <td>LU0 for the OS/400</td> </tr> <tr> <td>LU62</td> <td>LU6.2</td> </tr> <tr> <td>TCP</td> <td>TCP/IP</td> </tr> </tbody> </table> <p>The following values are valid for the session type parameter:</p> <ul style="list-style-type: none"> ◆ For MVS, VSE/ESA, or MS-DOS specify LU62 or SNA (SNA is the default). ◆ For AS/400, specify SNUF or LU62 <p>security node type classifies the node as an internal or external node. The fifth positional parameter is optional and provided with the trusted node security.</p> <p>EXTERNAL EXT specifies an external security classification for this node.</p> <p>INTERNAL INT specifies an internal security classification for this node.</p>	Session Type	Session Protocol	SNA	LU0	SNUF	LU0 for the OS/400	LU62	LU6.2	TCP	TCP/IP
Session Type	Session Protocol										
SNA	LU0										
SNUF	LU0 for the OS/400										
LU62	LU6.2										
TCP	TCP/IP										

Parameter	Description
ADJACENT.NODE (continued)	<p>data direction restriction identifies the copy initiation abilities of this adjacent node with the local node. The sixth positional parameter is the data direction restriction.</p> <p>This parameter is optional. Valid data direction values are:</p> <p>RECEIVE RECV indicates this node can receive data from the local node.</p> <p>SEND indicates this node can send data to the local node.</p> <p>BOTH indicates this node can send data to and receive data from the local node.</p> <p>NONE indicates this node can neither send data to nor receive data from the local node.</p>

Keyword Parameters

The following keyword parameters are for the network map adjacent node entry.

Parameter	Description
PARSESS=(max, default)	<p>specifies that parallel sessions can be used between a local node and an adjacent node. For parallel sessions support you must specify definitions for both nodes in the VTAM application. If you specify PARSESS, you must specify both a maximum value (max) and default value (default).</p> <p>If this parameter is not coded, the node is not considered parallel-session capable; the max and default values are set to 1.</p> <p>This parameter is required if you do PNODE=SNODE processing. It governs the number of simultaneous connections.</p> <p>max specifies the maximum number of sessions that can be established simultaneously between the local node and the adjacent node. The valid range for this subparameter is 2-28. Leave this field blank if parallel sessions are not available.</p> <p>Each session has a numeric class that determines which Processes are eligible to run on that session. See the Connect:Direct Processes Web site at www.sterlingcommerce.com/Documentation/processes/processhome.html for information about specifying the CLASS keyword.</p> <p>This value is ignored for transfers involving Connect:Direct for OpenVMS software. Instead, the maximum number of sessions is the sum of all lunames specified in the PNODE.LUS and SNODE.LUS parameters.</p> <p>default specifies the class to be assigned to a Process if one is not specified on the PROCESS statement or at submit time.</p> <p>See the Connect:Direct Processes Web site at www.sterlingcommerce.com/Documentation/processes/processhome.html for information on Process class and the <i>Connect:Direct for VSE/ESA Release Notes</i> chapter of the <i>Connect:Direct for VSE/ESA User's Guide</i> for further information about parallel sessions.</p> <p>If PARSESS is not specified, all Processes will be class 0.</p>

Parameter	Description
ENVIRONMENT=operating environment	<p>identifies the operating system environment of the adjacent node. This parameter is required only when the session type positional parameter (fourth) specifies LU6.2 protocol for OS/400 systems. This parameter can be used for documentation purposes for other protocols. The operating environment variable is limited to 6 characters.</p> <p>Valid values are: OS400, MVS, VM, VSE, TANDEM, VMS, VAX, OS2, MSDOS, MSP, NW, NT, PC, ST-VOS, and UNIX.</p>
LOGMODE=logmode entry name	<p>identifies the VTAM logmode entry defining the session protocol to be used when communicating with this node. <i>This parameter is required for LU6.2 nodes only.</i> This parameter is not valid for TCP/IP nodes.</p> <p>This parameter is optional for LU0 connections. If you specify the LOGMODE parameter for LU0 connections, it indicates that the RUSIZE defined within this LOGMODE is used for any transfer with this node. For a host-to-host transfer, the LOGMODE entry in the VTAM MODETAB of the SNODE is used to determine the RUSIZE. For a host-to-PC transfer, the LOGMODE entry in the host VTAM MODETAB is used.</p>
APPLIDS=(vtam applid1 [,vtam applid2,...])	<p>specifies a list of VTAM APPLID(s) that establish a session between the local Connect:Direct API and the remote Connect:Direct DTF. APPLIDs are defined on the local node. This parameter is not valid for TCP/IP. It is valid for MVS, MSP, VSE/ESA, and VM nodes only.</p> <p>These APPLIDs are not used for DTF-to-DTF communications. Therefore, these APPLIDs must be defined to the <i>local</i> VTAM and should not exist anywhere on the VTAM system for the adjacent node.</p>
NETID=networkid	<p>specifies the 1-8 character network ID for this node. At Process initiation, the network ID provided at session establishment is verified against the network ID specified on the network map entry for the adjacent node. If they do not match, the Process is terminated with an error.</p> <p>For multiple session signons, the network ID for the node signing on is verified against the network ID coded in the network map for the node being signed on to. If they do not match, the signon fails.</p> <p>If this keyword is not coded or the Connect:Direct initialization parameter NETMAP.CHECK is set to NO, the network ID will not be checked at Process initiation or multiple session signon.</p>
PNODE.LUS=(luname1 [,luname2,...])	<p>specifies the logical units used by a remote Connect:Direct node to initiate a session with this local node. Do not specify the communications name when you use this parameter. When the adjacent node initiates an operation, Connect:Direct for VSE/ESA will verify that the LU exists in the PNODE.LUS list.</p> <p>This parameter applies to OpenVMS nodes and OS/2 or NetWare LU6.2 dependent LUs.</p>

Parameter	Description
SNODE.LUS=(luname1 [,luname2,...]))	<p>(For OpenVMS) specifies the logical unit names used by the local node to initiate a session with this remote node. For all other platforms, it specifies the logical units used for all communications with the remote node.</p> <p>Communications to nodes not capable of parallel sessions may require the use of a pool of logical units. If an adjacent node is defined in its host environment to use more than one logical unit for communications, then each of the logical unit names that can be used to communicate with the local node must be defined to the local node on the corresponding adjacent node network map entry.</p> <p>TANDEM and OS/400 adjacent node definitions use the SNODE.LUS keyword only to define the LU pool.</p> <p>OpenVMS nodes assign the logical units in the pool to be either ACTIVE, session initiating, PASSIVE, or listening for session requests. This distinction in function is defined to the VSE/ESA node by specifying the ACTIVE logical units with the PNODE.LUS keyword and the PASSIVE logical units with the SNODE.LUS keyword.</p> <p>This parameter also applies to OS/2 and NetWare LU6.2 dependent LUs if you do not specify the LU name as the communications name.</p>
NDMPACE=(#sends, time)	<p>controls pacing when sending data. Use this parameter for TCP/IP nodes only.</p> <p>#sends indicates how many sends to issue before a response is required from the receiver. It is a numeric value between 0-1024. If 0 is coded, no response is required. The default is 0.</p> <p>time indicates the delay in milliseconds between sending of data blocks. It is a numeric value between 0-1024. If 0 is coded, no delay is used. The default is 0.</p>

Sample Connect:Direct Network Map Entries

The APPLIDs listed in the "APPLID=" pool can be used to sign on to the local node as well as the adjacent nodes in cross-domain logons.

```

LOCAL.NODE=( (CD.NODE.A  CDAPPL1  ,, $PW)  -
              TCQ=(TCX.FILE  TCQ.FILE) )
* PNODE=SNODE DEFINITION
ADJACENT.NODE=( (CD.NODE.A  CDAPPL1A)  -
                APPLIDS=(NAI01,NAI02,NAI03)
* SNA CONNECTIONS
ADJACENT.NODE=( (CD.NODE.B  CDAPPL2)  -
                APPLIDS=(NAI01,NAI02,NAI03)  *

```

Note: The LOCAL.NODE is also defined as an ADJACENT.NODE. This definition allows Processes to designate PNODE and SNODE as the same node.

PNODE=SNODE Processing

Connect:Direct has the ability to initiate Processes in which the local node is defined as both the initiating and target nodes (PNODE and SNODE). PNODE=SNODE processing is enabled by the insertion of an adjacent node entry in the network map where the node name is the same as that defined for the local node entry. Observe the following rules when setting up the adjacent node:

- ◆ Define an LU0 VTAM APPLID for the PNODE=SNODE network map entry. The APPLID of the adjacent node entry must not specify an LU6.2 logmode entry name.
- ◆ The communications name for the adjacent node must be different than the communications name of the local node. If the names are the same, PNODE=SNODE processing is disabled at installation.
- ◆ Code the PARSESS=(max, default) parameter in the network map adjacent node entry to govern the number of simultaneous PNODE=SNODE connections. The PARSESS parameter is described on page 69.

Example Node Records

This section contains examples of local and adjacent node records for various platforms.

Local Node and Adjacent Node Record for the Local Node

The following is an example of a LOCAL.NODE and ADJACENT.NODE record for the LOCAL.NODE.

```

LOCAL.NODE=((CD.DALLAS , CDDD1 , , XYZZY)                                -
            TCQ=(DSC.DALLAS.TCX DSC.DALLAS.TCQ))
/* THE FOLLOWING ENTRY IS FOR THE LOCAL NODE */
ADJACENT.NODE=(PARSESS=(12 2) (CD.DALLAS CDDD2)                       -
              APPLIDS=(CDDI2  CDDI3  CDDI11  CDDI12                   -
                      CDDI17  CDDI18  CDDI32  CDDI41  CDDI42))

```

VSE or MVS SNA LU0 Adjacent Node Example

The example which follows shows an adjacent MVS node named CD.NYCVSE with the communications name (VTAM APPLID) of CDDD10. The APPLIDS parameter indicates nine API sessions are possible.

```

ADJACENT.NODE=(PARSESS=(4 2) (CD.NYCVSE,CDDD10)                       -
              APPLIDS=(CDAPI01  CDAPI02  CDAPI03  CDAPI04             -
                      CDAPI05  CDAPI06  CDAPI07  CDAPI08  CDAPI09))

```


VSE or MVS TCP/IP Adjacent Node Examples

The example which follows shows three adjacent node definitions with session protocol types of TCP/IP.

- ◆ The first, with a TCP net name of VSE.CD.CHICAGO, specifies the default TCP/IP port number by leaving the communications name positional parameter null.
- ◆ The second, with a TCP/IP net name of VSE.CD.DALLAS, specifies a TCP/IP port number of 4444.
- ◆ The third, with a TCP/IP net name of VSE.CD.AUSTIN, specifies a TCP/IP port number of 4443 and an IP address of 199.8.8.8.

```

ADJACENT.NODE=( PARSESS=( 4, 2)           -
                (VSE.CD.CHICAGO, ,199.1.4.51,TCP) -
                ENVIRONMENT=VSE)
ADJACENT.NODE=( PARSESS=( 4, 2)           -
                (VSE.CD.DALLAS, 4444, ,TCP) -
                ENVIRONMENT=VSE)
ADJACENT.NODE=( PARSESS=( 4, 2)           -
                (VSE.CD.AUSTIN, 4443, 199.8.8.8, TCP) -
                ENVIRONMENT=VSE)

```

VSE or MVS SNA LU6.2 Adjacent Node Example

The following example shows an adjacent node definition for a node named CD.LAVSE with a communications name (applid) of APPLLAI and a session protocol type of LU6.2. The operating environment of this adjacent node is VSE, and the VTAM logmode entry which defines the session protocol to be used when communicating with this node is LU62MOD4.

```

ADJACENT.NODE=( PARSESS=( 4, 2)           -
                (CD.LAVSE, APPLLAI, , LU62) -
                ENVIRONMENT=VSE LOGMODE=LU62MOD4 -
                APPLIDS=( CDDD2, CDDD3, CDDD4) )

```

The LOGMODE parameter is required for LU6.2.

VSE or MVS Trusted Node Example

The following example shows an adjacent node definition for a node named SC.NODE.A with a security type of external (EXT) and data direction restriction of SEND.

```

ADJACENT.NODE=( PARSESS=( 4, 2)           -
                (SC.NODE.A, NVSED20, , , EXT, SEND) -
                APPLIDS=( NVSEA36, NVSEA37, NVSEA38) )

```

OpenVMS LU0 Adjacent Node Example

The following is a sample adjacent node for Connect:Direct for OpenVMS.

```

ADJACENT.NODE=( ( CD.DALLAS.VMS)                -
PARSESS=(9)                                     -
PNODE.LUS=(N91LU09 N91LU0A  N91LU0B  N91LU0C   -
            N91LU0C N91LU0D  N91LU0E  N91LU0F   -
            N91LU10)                             -
SNODE.LUS=(N91LU07  N91LU08))

```

MS-DOS LU0 Adjacent Node Examples

The following example shows adjacent MS-DOS nodes and their communications names (LUs).

```

ADJACENT.NODE=(N21.02,N21LU02)
ADJACENT.NODE=(N21.03,N21LU02)
ADJACENT.NODE=(N21.04,N21LU02)
ADJACENT.NODE=(CD.ATLANTAPC,CDLU20)

```

MS-DOS LU6.2 Adjacent Node Example

The following example shows an adjacent MS-DOS node with a session protocol type of LU6.2, and environment of PC, and logmode entry name of NDM62PC.

```

ADJACENT.NODE=( (M92.02,M592LU02,,LU62)        -
ENVIRONMENT=MSDOS LOGMODE=NDM62PC)

```

The LOGMODE parameter is required for LU6.2.

NetWare LU6.2 Adjacent Node with Independent LU Name Designated

The following example shows a designated LU name. Enter the LU name (NWL01) in the Communications Name field. You must specify LU62 as the session type. The LOGMODE parameter is required for LU 6.2 protocol.

```

ADJACENT.NODE=(PARSESS=(6, 2)                  -
                (NW.LU62.DALLAS, NWLU01, , LU62) -
                ENVIRONMENT=NW, LOGMODE=NDM621K)

```

NetWare LU6.2 Adjacent Node with Dependent LUs

The following example shows the use of the SNODE.LUS parameter that allows multiple sessions between the two nodes using dependent LUs. The session type must be LU62. The LOGMODE parameter is required for LU 6.2 protocol.

```

ADJACENT.NODE=(PARSESS=(4, 2) -
(NW.CD.NY,, , LU62) -
ENVIRONMENT=NW, LOGMODE=LU62MOD3 -
SNODE.LUS=(NYLU01, NYLU02, NYLU03, NYLU04) )

```

NetWare LU6.2 Adjacent Node with Independent and Dependent LUs

The following example shows an adjacent Netware node with a session protocol type of LU6.2 and a logmode entry name of LU62MOD3. Specifying both the PNODE.LUS and the SNODE.LUS parameters provides a list of LUs (dependent or independent) that can be used to communicate.

```

ADJACENT.NODE=( ( CD.DALLAS.NW, , , LU62) -
ENVIRONMENT=NW, LOGMODE=LU62MOD3 -
PARSESS=(9) -
PNODE.LUS=(N91LU09 N91LU0A N91LU0B N91LU0C N91LU0C -
N91LU0D N91LU0E N91LU0F N91LU10) -
SNODE.LUS=(N91LU07 N91LU08) )

```

NetWare TCP/IP Adjacent Node Example

The example which follows shows two adjacent node definitions with session protocol types of TCP/IP.

- ◆ The first, with the TCP net name of NW.CD.CHICAGO, specifies the default TCP/IP port number by leaving the communications name positional parameter null and specifies an IP address of 199.4.4.4.
- ◆ The second, with the TCP net name of NW.CD.DALLAS, specifies a TCP/IP port number of 5555 and an IP address of 199.5.5.5.

```

ADJACENT.NODE=(PARSESS=(6, 2) -
(NW.CD.CHICAGO, , 199.4.4.4, TCP) -
ENVIRONMENT=NW)
ADJACENT.NODE=(PARSESS=(6, 2) -
(NW.CD.DALLAS, 5555, 199.5.5.5, TCP) -
ENVIRONMENT=NETWARE)

```

Notice that there are no APPLID or LOGMODE keywords used for any TCP/IP node. A warning is generated for any unneeded keyword or subparameter, and the coded value is ignored.

OS/2 LU0 Adjacent Node Examples

The following example shows adjacent OS/2 nodes and their communications names (applids).

```
ADJACENT.NODE=(M83.02,M58CLU03)
ADJACENT.NODE=(SACTEST2,M596LU04)
```

OS/2 LU6.2 Adjacent Node with Independent LU Example

The following example shows an adjacent node named OS2.CD.LA with an independent LU communications name of APPLLA1, session protocol type of LU6.2, and a logmode entry name of LU62MOD2.

```
ADJACENT.NODE=(PARSESS=(6, 2) -
(OS2.CD.LA,APPLLA1,CDOS2,LU62) -
ENVIRONMENT=OS2 LOGMODE=LU62MOD2)
```

The ENVIRONMENT=OS2 parameter is required for OS/2 nodes using the LU6.2 protocol. The LOGMODE parameter is required for LU6.2 protocol. Notice that the third positional parameter is required to be CDOS2.

OS/2 LU6.2 Adjacent Node with Dependent LU Example

The following example shows an adjacent OS/2 node named OS2.CD.NY with a session protocol type of LU6.2 and a logmode entry name of LU62MOD3. The SNODE=LUS parameter defines the dependent LU pool.

```
ADJACENT.NODE=(PARSESS=(4, 2) -
(OS2.CD.NY,,CDOS2,LU62) -
ENVIRONMENT=OS2 LOGMODE=LU62MOD3 -
SNODE.LUS=(NYLU01,NYLU02,NYLU03,NYLU04))
```

The ENVIRONMENT=S2 parameter is required for OS/2 nodes using the LU6.2 protocol. The LOGMODE parameter is required for LU6.2 protocol. The third positional parameter *must* be CDOS2.

OS/400 SNUF Adjacent Node Example

The following example shows an adjacent node named AS400.CD.TX with a remote library name of LBNAME and session protocol type of LU0 (SNUF). The SNODE.LUS parameter defines the dependent LU pool.

```
ADJACENT.NODE=(PARSESS=(4, 2) -
(AS400.CD.TX,,LBNAME,SNUF) -
SNODE.LUS=(N11LU01,N11LU02,N11LU03,N11LU04))
```

OS/400 (LU6.2) Adjacent Node with Independent LU Example

The following example shows an adjacent node named AS400.CD.LA with an independent LU communications name of APPLLA1, a remote library name of CDLIB1, session protocol type of LU6.2, and a logmode entry name of LU62MOD2.

```
ADJACENT.NODE=( PARSESS=( 6, 2)           -
                (AS400.CD.LA, APPLLA1, CDLIB1, LU62) -
                ENVIRONMENT=OS400 LOGMODE=LU62MOD2)
```

The ENVIRONMENT=OS400 parameter is required for OS/400 nodes using the LU6.2 protocol. The LOGMODE parameter is required for LU6.2 protocol.

OS/400 (LU6.2) Adjacent Node with Dependent LU Example

The following example shows an adjacent OS/400 node named AS400.CD.NY with a remote library name of CDLIB1, a session protocol type of LU6.2, and a logmode entry name of LU62MOD3. The SNODE=LUS parameter defines the dependent LU pool.

```
ADJACENT.NODE=( PARSESS=( 4, 2)           -
                (AS400.CD.NY, , CDLIB1, LU62) -
                ENVIRONMENT=OS400 LOGMODE=LU62MOD3 -
                SNODE.LUS=( NYLU01, NYLU02, NYLU03, NYLU04))
```

The ENVIRONMENT=AS400 parameter is required for OS/400 nodes using the LU6.2 protocol. The LOGMODE parameter is required for LU6.2 protocol.

Tandem LU0 Adjacent Node Example

The following is a sample adjacent node for Connect:Direct for Tandem.

```
ADJACENT.NODE=( ( CD.TANDEM)           -
                PARSESS=( 2)           -
                SNODE.LUS=( N81LU0C N81LU0D))
```

UNIX TCP/IP Adjacent Node Example

The example which follows shows two adjacent node definitions with session protocol types of TCP/IP.

- ◆ The first, with the TCP net name of UNIX.CD.CHICAGO, specifies the default TCP/IP port number by leaving the communications name positional parameter null.
- ◆ The second, with the TCP net name of UNIX.CD.DALLAS, specifies a TCP/IP port number of 5555 and an IP address of 199.5.5.5.

```

ADJACENT.NODE=( PARSESS=( 6, 2) -
                (UNIX.CD.CHICAGO,,TCP) -
                ENVIRONMENT=UNIX)
ADJACENT.NODE=( PARSESS=( 6, 2) -
                (UNIX.CD.DALLAS,5555,199.5.5.5,TCP) -
                ENVIRONMENT=UNIX)

```

Notice that there are no APPLID or LOGMODE keywords used for any TCP/IP node. A warning is generated for any unneeded keyword or subparameter, and the coded value is ignored.

UNIX LU6.2 Adjacent Node Example

The following example shows an adjacent UNIX node with an communications name (applid) of D1UNIX and a session protocol type of LU6.2. The logmode entry name is LU62MODE.

```

ADJACENT.NODE=( PARSESS=( 6, 2) -
                (UNIX.lu62.DALLAS,D1UNIX,,LU62) -
                LOGMODE=LU62MODE -
                ENVIRONMENT=UNIX)

```

The LOGMODE parameter is required for LU6.2.

VM SNA LU0 Adjacent Node Example

The following example shows an adjacent node named CD.BOSTONVM with a communications name (applid) of CDDD16.

```

ADJACENT.NODE=( PARSESS=( 4 2) (CD.BOSTONVM,CDDD16) -
                APPLIDS=(CDAPI01 CDAPI02 CDAPI03 CDAPI0 -
                CDAPI05 CDAPI06 CDAPI07 CDAPI08 CDAPI09)) -

```

VSE SNA LU0 Adjacent Node Example

The following example shows an adjacent node named CD.DALLASVSE with a communications name (applid) of CDDD22.

```

ADJACENT.NODE=( PARSESS=( 6 2) -
                CD.DALLASVSE,CDDD22) -
                APPLIDS=(CDAPI01 CDAPI02 CDAPI03 CDAPI04 CDAPI05))

```

VSE LU6.2 Adjacent Node Example

The following is a sample Connect:Direct for VSE/ESA (LU6.2) adjacent node.

```
ADJACENT.NODE = (PARSESS = (4, 2) -
(CD.LAVSE,APPLLAI, ,LU62) -
ENVIRONMENT=VSE LOGMODE=LU62MOD4 -
APPLIDS=(CDDD2,CDDD3,CDDD4) )
```

Windows NT TCP/IP Adjacent Node Example

The example which follows shows two adjacent node definitions with session protocol types of TCP/IP.

- ◆ The first, with the TCP net name of NT.CD.CHICAGO, specifies the default TCP/IP port number by leaving the communications name positional parameter null and specifies an IP address of 199.4.4.4.
- ◆ The second, with the TCP net name of NT.CD.DALLAS, specifies a TCP/IP port number of 5555 and an IP address of 199.5.5.5.

```
ADJACENT.NODE=(PARSESS=(6, 2) -
(NT.CD.CHICAGO, ,199.4.4.4,TCP) -
ENVIRONMENT=NT)
ADJACENT.NODE=(PARSESS=(6, 2) -
(NT.CD.DALLAS,5555,199.5.5.5,TCP) -
ENVIRONMENT=NT)
```

Notice that there are no APPLID or LOGMODE keywords used for any TCP/IP node. A warning is generated for any unneeded keyword or subparameter, and the coded value is ignored.

Updating the Network Map

The network map is initially created in step 6 of the installation procedure when the network map source is the input to the network map load program. The contents of the network map source define what the network map will contain.

The network map source contains one local node entry and multiple adjacent node entries. It can contain \$\$ACTION verbs added during previous maintenance. See *Using \$\$ACTION VERBS* on page 81 for how to use \$\$ACTION verbs to update your network map source.

You can update the network map source while Connect:Direct is not executing or dynamically while Connect:Direct is executing.

Updating While Connect:Direct is Not Executing

You can update the network map using the network map source and the JNETLOAD program which loaded the source at initialization.

Note: Local node updates can only be accomplished using this method.

Perform the following steps:

1. Change the network map source.
Loaded at installation as NETMAP01.S
2. Stop Connect:Direct.
3. Delete and redefine the network map.
Refer to JCL in JNETDEF.S
4. Reload the network map.
Refer to JCL in JNETLOAD.S
5. Restart Connect:Direct.

Updating the Network Map Dynamically

The network map can also be updated without deleting and redefining it. You can update the network map source without stopping Connect:Direct by using the UPDATE NETMAP command. As with most commands, you can execute the command through a batch job or through the console. Both methods use \$\$ACTION verbs as part of the network map source.

Note: This method of updating the network map is only available for adjacent nodes.

The format of the UPDATE NETMAP command is as follows:

Label	Command	Parameters
(optional)	UPDate NETMAP	WHERE (NETIN put=filetype(member name) NETLOG=[DLBLname NONE]) DIS PRT

WHERE is the only required parameter.

Parameter	Description
WHERE (NETINput = filename (member name) NETLOG = [DLBLname NONE])	<p>specifies the network map source file and where the update activity is to be reported.</p> <p>NETINput = filetype (member name) specifies the name of the network map source member. The member must be cataloged into a sublibrary contained in the <i>SOURCE</i> LIBDEF. The network map source can contain multiple basic action verbs, multiple special purpose action verbs, or a combination of both. Action verbs are described on page 81. The source used to update the network map can be the entire network map source or a subset of it.</p> <p>NETLOG = [DLBLname NONE] specifies where the update activity is to be reported. If the field is left blank the update activity will be reported to the NDMLOG data set. Regardless of which option is selected the activity is recorded in the Statistics file as <i>WTO</i> records.</p> <p>DLBLname specifies the data definition name allocated to the Connect:Direct DTF where the update activity is to be reported. NONE specifies that no update activity is reported.</p>
DIS PRT	<p>specifies the output destination.</p> <p>DIS indicates that the activity is to be reported in display format, either to the console for console requests or to the DLBLname for batch requests.</p> <p>PRT indicates that the output is to be routed to SYSOUT for batch requests.</p>

Using \$\$ACTION VERBS

Add \$\$ACTION verbs to the network map source described in the NETINPUT parameter of the UPDATE NETMAP command. Each verb defines the action to take for the node entry immediately following the action verb. There are three basic action verbs and three special purpose action verb pairs.

Basic Action Verbs

The three basic action verbs are:

Verb	Description
\$\$INSERT	insert the following node into the network map.

Verb	Description
\$\$UPDATE	update the following existing network map node entry. Node entry updates are performed as a replacement at the keyword level; therefore, updates of list-type keywords, like APPLIDS=, require the entire list to be specified.
\$\$DELETE	delete the following existing network map node entry.

Special Purpose Verb Pairs

The three special purpose verb pairs are:

To perform syntax checking:

Verb	Description
\$\$SYNTAX	perform a syntax check of the network map control statement(s) following this verb.
\$\$ENDSYNTAX	terminate syntax checking.

To perform verification of node definition:

Verb	Description
\$\$VERIFY	verify that the node definition(s) following this verb match those in the network map.
\$\$ENDVERIFY	terminate the verification action.

To perform the basic action for a block of node entries:

Verb	Description
\$\$BLKxxxxxx	perform the basic action verb defined by xxxxxx for the block of node entries following this verb. Replace xxxxxx with either INSERT, UPDATE, or DELETE.
\$\$ENDxxxxxx	terminate the requested block action.

Update Using the Connect:Direct Batch Utility

One way to issue the UPDATE NETMAP command is through the Connect:Direct batch utility. To update the network map using this method, perform the following steps:

1. Change the network map source using the \$\$ACTION verbs.
2. Place the UPDATE NETMAP commands in the DMBATCH job stream as presented in the *Connect:Direct for VSE/ESA User's Guide*.

3. Ensure that Connect:Direct is running.
4. Submit the job.
5. Correct any errors identified on the activity report and resubmit if necessary.
6. Verify the results.

\$\$ACTION Verb Examples

The following are examples of updating the network map through the use of action verbs.

\$\$INSERT Example

The following \$\$INSERT command inserts an adjacent node into the network map.

```

$$INSERT
  ADJACENT.NODE= ( ( CD.NODE2  APPLID2  )      -
                   PARSESS=( 5, 2)           -
                   APPLIDS=(RAPPL1) )

```

The output appears as follows:

```

= = > * * * START NETMAP UPDATE          * * *
= = >          DATE: 02/27/1997  TIME=14:59:26
= = >          SMUPNLGI NETLOG=NONE REQUIRED, LOGGING INACTIVE
=====
*INSERT THE FOLLOWING NODE DEFINITION
$$INSERT
  ADJACENT.NODE= ( ( CD.NODE2  APPLID2  )      -
                   PARSESS=( 5, 2)           -
                   APPLIDS=(RAPPL1  ) )
= = >          SMUP032I APPLIDS RECORD INSERTED
= = >          SMUP034I ADJACENT.NODE RECORD INSERTED
= = >          SMUP008I REQUEST SUCCESSFUL FOR NODE=CD.NODE2
=====

```

The first message shows that logging is not requested; therefore, no record is kept of the transaction (except in the statistics file). The last messages indicate that the information for the specified adjacent.node is successfully inserted.

\$\$UPDATE Example

The following \$\$UPDATE command updates an adjacent node in the network map by adding the RAPPL2 applid and changing the maximum parallel sessions to four.

```

$$UPDATE
  ADJACENT.NODE= ( ( CD.NODE2  APPLID2  )      -
                   PARSESS=( 4, 2)           -
                   APPLIDS=(RAPPL1  RAPPL2  ) )

```

The output appears as follows:

```

= = > * * * START NETMAP UPDATE          * * *
= = >          DATE: 02/27/1997  TIME=14:59:26
= = >          SMUPNLGI LOGGING ACTIVE - LOG DLBLName=NDMLOG
=====
*UPDATE THE FOLLOWING NODE DEFINITION ADDING RAPPL2
*CHANGING MAXIMUM PARALLEL SESSIONS TO FOUR (4).
$$UPDATE
  ADJACENT.NODE=( ( CD.NODE2  APPLID2 )          -
                    PARSESS=(4,2)                -
                    APPLIDS=(RAPPL1 RAPPL2) )
= = >          SMUP032I APPLIDS RECORD UPDATED
= = >          SMUP034I ADJACENT.NODE RECORD UPDATED
= = >          SMUP008I REQUEST SUCCESSFUL FOR NODE=CD.NODE2
=====

```

The first message shows that logging is requested and that a record of the transaction will be recorded in NDMLOG. The last messages indicate that the adjacent.node information is successfully updated.

\$\$DELETE Example

The following \$\$DELETE command will delete an adjacent node from the network map.

```

$$DELETE
  ADJACENT.NODE=( (CD.NODE2  APPLID2)          -
                    PARSESS=(4,2)                -
                    APPLIDS=(RAPPL1 RAPPL2) )

```

The output appears as follows:

```

= = > * * * START NETMAP UPDATE          * * *
= = >          DATE: 02/27/1997  TIME=15:09:36
= = >          SMUPNLGI NETLOG=NONE REQUIRED, LOGGING INACTIVE
=====
$$DELETE
  ADJACENT.NODE=( (CD.NODE2  APPLID2)          -
                    PARSESS=(4,2)                -
                    APPLIDS=(RAPPL1 RAPPL2 ) )
= = >          SMUP032I APPLIDS RECORD DELETED
= = >          SMUP034I ADJACENT.NODE RECORD DELETED
= = >          SMUP008I REQUEST SUCCESSFUL FOR NODE=CD.NODE1
=====

```

The first message indicates that logging is requested; therefore, no record is kept of the transaction. The last messages indicate that the applids and adjacent node records are successfully deleted.

\$\$\$SYNTAX Example

The following \$\$\$SYNTAX command performs a syntax check on the specified nodes.

```

$$$SYNTAX
  LOCAL.NODE=((CD.NODE1 APPLID1 ,, SUPERUSR) -
              TCQ=( TCQ TCX ))
  ADJACENT.NODE=((CD.NODE1 APPLID1) -
                PARSESS=(5,2) -
                APPLIDS=(LAPPL1 LAPPL2 LAPPL3))
  ADJACENT.NODE=((CD.NODE2 APPLID2) -
                PARSESS=(5,2) -
                APPLIDS=(RAPPL1 ))
$$$ENDSYNTAX

```

The output appears as follows:

```

=====
= = > * * *          START NETMAP UPDATE          * * *
= = >          DATE: 02/27/1997  TIME=13:49:16
(1)  = = >          SMUPNLGI NETLOG=NONE REQUIRED, LOGGING INACTIVE
=====
      $$$SYNTAX
(2)  = = > SMUP011I 'SYNTAX ' ACTION STARTED
=====
      LOCAL.NODE=(( CD.NODE1 APPLID1 ,, SUPERUSR) -
                  TCQ=( TCQ TCX ))
(3)  = = > SMUP005I LOCAL.NODE RECORD PROCESSING NOT ALLOWED
      BYPASSED
=====
      ADJACENT.NODE=(( CD.NODE1 APPLID1 ) -
                    PARSESS=(5,2) -
                    APPLIDS=(LAPPL1 LAPPL2 LAPPL3))
(4)  = = > SMUP008I REQUEST SUCCESSFUL FOR NODE=CD. NODE1
=====
      ADJACENT.NODE=(( CD.NODE2 APPLID2 ) -
                    PARSESS=(5,2) -
                    APPLIDS=(RAPPL1))
(4)  = = > SMUP008I REQUEST SUCCESSFUL FOR NODE=CD. NODE2
=====
      $$$ENDSYNTAX
(5)  = = > SMUP012I 'SYNTAX ' ACTION STOPPED
=====

```

The messages are numbered in the example for clarification; they are not numbered on the actual output.

1. Logging is not requested, so no transaction record is kept.
2. Syntax check of network map control statements starts.
3. No processing is allowed against the local.node record.
4. Requests for syntax checking on nodes are successful.
5. Syntax checking completes.

\$\$VERIFY Example

The following \$\$VERIFY command verifies the definition of the specified adjacent node record prior to updating the network map.

```

$$VERIFY
ADJACENT.NODE=((CD.NODE2 APPLID2)      -
                PARSESS=(5,2)          -
                APPLIDS=(RAPPL1))
$$ENDVERIFY

```

The result is the following output:

```

      == > * * *          START NETMAP UPDATE          * * *
      == > DATE:  02/27/1997  TIME=15:35:16
(1)   == > SMUPNLGI NETLOG=NONE REQUIRED, LOGGING INACTIVE
=====
      $$VERIFY
(2)   == > SMUP011I 'VERIFY  ' ACTION STARTED
=====
      ADJACENT.NODE=(( CD.NODE2  APPLID2 ) -
      PARSESS=(5,2) -
      APPLIDS=(RAPPL1 ))
(3)   == > SMUP092I  APPLIDS RECORD DID NOT MATCH
      == > SMUP094I ADJACENT.NODE RECORD DID NOT MATCH
      == > SMUP096I RECORDS DO NOT MATCH - VERIFICATION FAILED
                                FOR NODE
      =CD.NODE2
=====
      $$ENDVERIFY
(4)   == > SMUP012I 'VERIFY  ' ACTION STOPPED
=====

```

The messages are numbered in the example for clarification; they will not be numbered on the actual output.

1. Logging was not requested; therefore no record will be kept of the transaction.
2. Verification of the node definition to the network map file has started.
3. The applids and adjacent node records did not match the network map file definitions.
4. Verify has completed.

Planning and Controlling Security

Connect:Direct security can vary in different environments from no security support to a total security package controlling access to all data.

To meet security requirements in different environments, Connect:Direct provides a range of options. Some of these options are built into Connect:Direct itself and some are available from user-customized exit routines.

This chapter explains how security works with Connect:Direct and provides the information you need to plan and install Connect:Direct security in your particular environment. It includes discussions of the following:

- ◆ Planning for security
- ◆ Security exits invoked during processing
- ◆ Implementing security exits
- ◆ Connect:Direct authorization facility

Planning for Security

Connect:Direct supports signon security checking through the Connect:Direct authorization facility.

Levels of Security Support

Connect:Direct security support includes but is not limited to:

- ◆ Files
- ◆ Jobstreams (RUN JOB Exit)
- ◆ Application programs (RUN TASK Exit)
- ◆ Users (Connect:Direct Authorization Facility)
- ◆ Connect:Direct functions (Connect:Direct Authorization Facility)

SECURITY Initialization Parameter

Connect:Direct provides the SECURITY initialization parameter so that a stage 2 security exit can be specified. This exit is invoked during processing of the Signon command and Process start and data set access. With a stage 2 security exit, when a request is made to Connect:Direct for signon or file access, the request is passed directly to the security exit for authorization checking.

Note: High-level assembler for MVS & VM & VSE is required to assemble the sample security exits.

Authorization Facility

For installations with no security packages installed, the Connect:Direct authorization facility can be used. If the SECURITY.EXIT initialization parameter is not specified, or is commented out, the Connect:Direct authorization facility will be used for signon security and assigning functional authority within Connect:Direct. The Connect:Direct authorization facility provides no data set access security checking. Refer to *Maintaining the User Authorization File* on page 45 for more information about user authorization.

Security Exits Invoked During Processing

Before you can begin to plan implementation of Connect:Direct security, you should understand your host security environment and security system. You should also understand the flow of work through the Connect:Direct system and the processing flow when security controls are invoked.

There are four Connect:Direct security exits:

- ◆ Stage 1 signon security exit
- ◆ Stage 2 security exit
- ◆ Run Job security exit
- ◆ Run Task security exit

Connect:Direct has two major processing flows through which these security exits are invoked. They are the SIGNON command sequence and the Process execution sequence.

The **SIGNON command sequence** is the first flow through which a Connect:Direct terminal user, console operator, or batch application gains access to Connect:Direct functions. One or more of the following control points is invoked:

- ◆ Stage 1 signon security exit
- ◆ Connect:Direct Authorization Facility
- ◆ Stage 2 security exit

The **Process execution sequence** is the second flow through which Connect:Direct services execute on behalf of a user request. One or more of the following control points is invoked:

- ◆ Process start invokes the stage 2 security exit
- ◆ Connect:Direct Copy statement invokes the stage 2 security exit

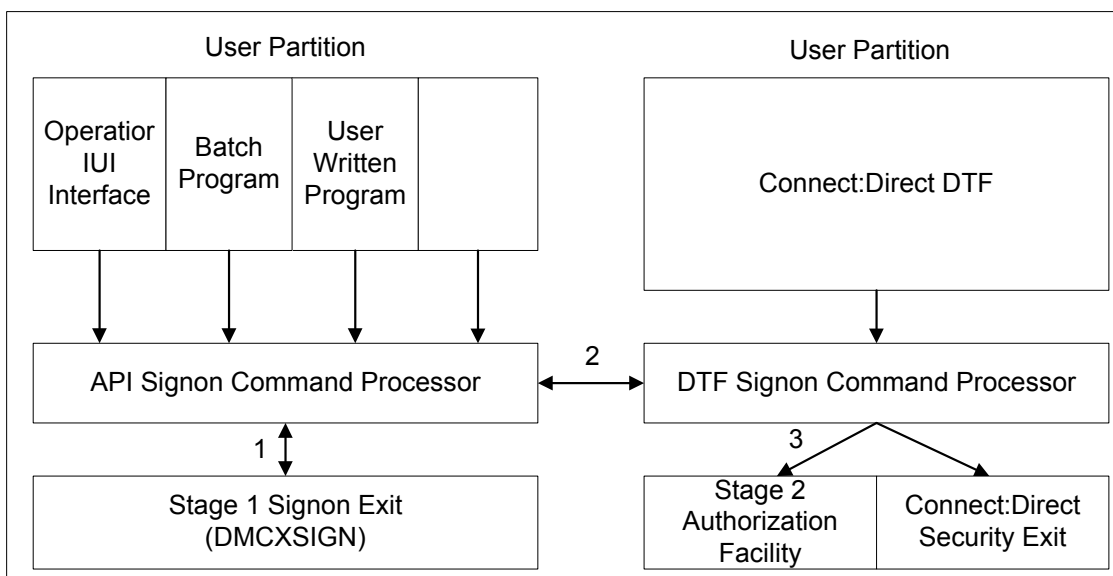
- ◆ Connect:Direct Run Task statement invokes the Run Task security exit
- ◆ Process end invokes the stage 2 security exit
- ◆ Connect:Direct Run Job statement invokes the stage 2 security exit and/or the Run Job security exit

These two processing flows during which security exits are invoked are described in the following sections.

Security During Signon Command

When you execute a SIGNON command (either through the batch, interactive, or operator interface), there are security control points in both the Connect:Direct user partition or API and the Connect:Direct DTF partition. As shown in the following figure, the initial control point is the stage 1 signon security exit. This control point is optional and is in the form of a user exit that gains control in the user's partition. The exit can inspect and modify the SIGNON command parameters.

The next control point occurs in the DTF partition and can take the form of a stage 2 security exit or the Connect:Direct authorization facility.



Refer to the numbers in the previous figure as you trace the SIGNON command flow described in the following paragraphs.

1. When you issue a Connect:Direct SIGNON command, the API SIGNON command processor calls the stage 1 signon exit. If the stage 1 exit is not found, normal signon processing continues.

When invoked, the stage 1 exit receives a pointer to the Connect:Direct User Interface Control Block (UICB) that contains information regarding the signon attempt. See page 146 for UICB field listings.

If you specified a password on the SIGNON command, the stage 1 exit returns control to Connect:Direct without making any modifications to the UICB, and the signon processing proceeds. In this case, the stage 2 exit will verify the USERID and PASSWORD that were coded on the SIGNON command for system entry validation and all subsequent security calls.

If you did not specify a password on the SIGNON command, Connect:Direct will extract the USERID from the security system control block that has already been built for this partition (when the BATCH job began execution) and put that USERID into the UICB.

Note: It is important to remember that the stage 1 exit keys off the password, not the userid. So, if a password has not been specified but a userid has, the stage 1 exit will ignore that userid and overlay it with the userid that is picked up from the security system control block.

Once the userid has been moved to the UICB, the exit will fill in a special password of IUI, BATCH, or STC, depending upon what environment the signon came from (since Connect:Direct cannot access the password), and control is returned to Connect:Direct.

The benefit of running with a stage 1 signon exit is that Connect:Direct batch jobs need not have hard coded passwords in their SYSIN data streams.

Note: As the sample stage 1 exit is shipped, the dummy passwords of IUI, BATCH, and STC are coded in the exit. You should change these passwords for each installation, to avoid the chance that another site is using the same dummy passwords as your site. You can change these passwords by editing the source for DMCXSIGN and the appropriate validation in the macro DMGSECUR (for the stage 2 exit).

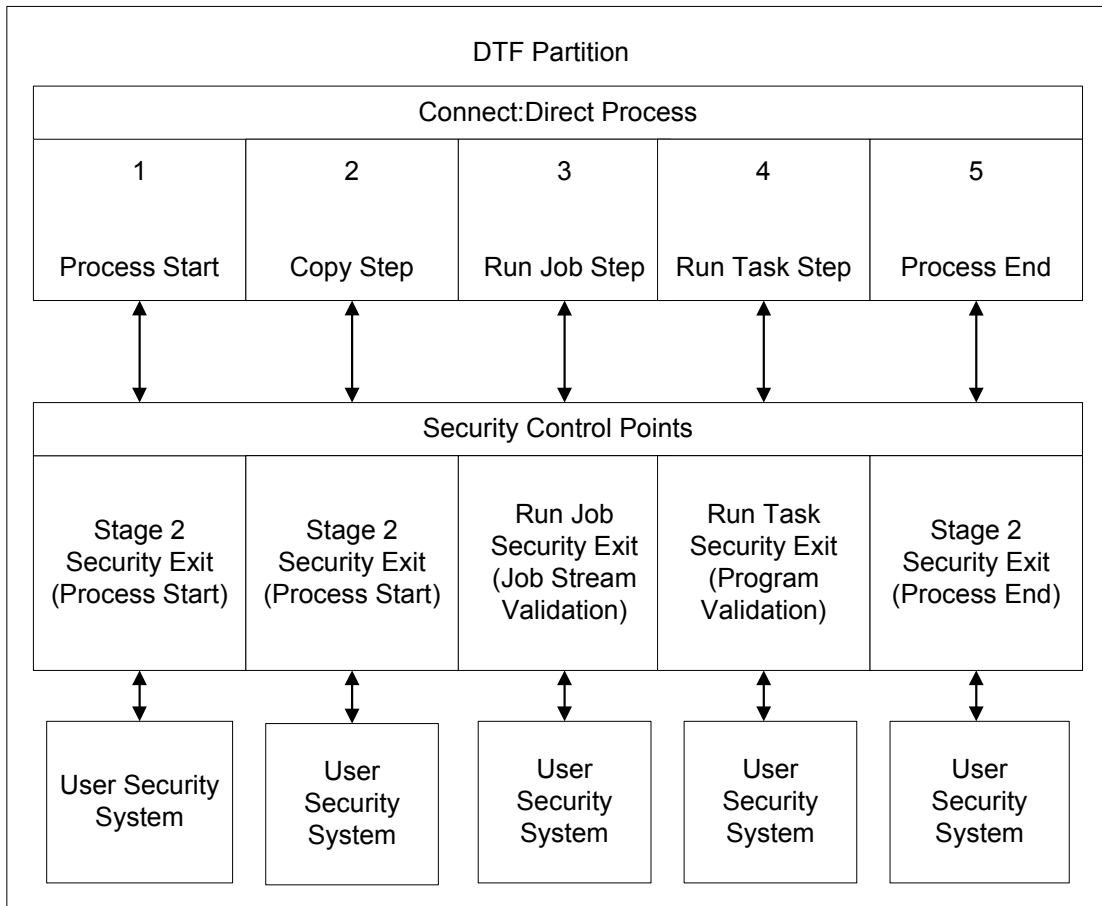
2. If the stage 1 processing is successful, the API SIGNON command processor passes the SIGNON command to the DTF where the DTF SIGNON command processor is invoked.
3. The DTF SIGNON command processor calls the stage 2 security exit or the Connect:Direct authorization facility.

The stage 2 exit will recognize special passwords of IUI, BATCH, and STC as having been assigned by the stage 1 exit, and all calls to the security system for verifications will verify authorizations by userid only.

Regardless of how your system is implemented, this processing flow verifies the authority of the requesting user to perform Connect:Direct functions by checking the ABM (Authorization Bit Mask) for this user. The ABM is built through the stage 2 security exit or through the Connect:Direct authorization facility at signon and Process start.

Security During Process Execution

When Connect:Direct executes a Process on behalf of a user, there are several security control points in the DTF as shown in the following figure.



Refer to the numbers in the previous figure as you trace the Process flow described in the following paragraphs.

1. **Process start**—This point in the stage 2 security exit gains control whenever a Process begins initial execution or restart execution and allows verification of the authority of the requesting user to perform the Connect:Direct functions contained in the Process.
2. **File access**—This point in the stage 2 security exit gains control during Process execution whenever a COPY or RUN JOB statement is encountered. It allows verification of the user's access to read or write the file defined in the COPY statement.

With the RUN JOB statement, the exit allows verification of the user's access to read the file containing the job stream to be submitted.

3. **Run Job**—This exit point allows for job stream validation and gains control when these conditions exist:
 - ◆ RUN JOB statement is encountered during Process execution
 - ◆ RUN.JOB.EXIT initialization parameter is specified

4. **Run Task**—This exit point allows for program validation and gains control when these conditions exist:
 - ◆ RUN TASK statement is encountered during Process execution
 - ◆ RUN.TASK.EXIT initialization parameter is specified
5. **Process end**—This point in the stage 2 security exit gains control whenever a Process terminates, whether normally or abnormally. This exit point assists in cleaning up the security resources involved in Process execution.

Note: Copy, Run Job, and Run Task exit functions are entered for every occurrence of the associated statement in a Connect:Direct Process.

Implementing Security Exits

Connect:Direct provides a number of options for implementing security. The choice is generally dictated by the environment in which Connect:Direct is being installed. Since some options will only work in certain environments and other options are mutually exclusive, each of the control points described earlier is listed here with the applicable environments and the implementation details.

Stage 1 Signon Security Exit

This control point allows verification of the format and contents of the SIGNON command. The following requirements and restrictions apply:

- ◆ The stage 1 signon exit is implemented as an executable module.
- ◆ The name of the module must be DMCXSIGN.
- ◆ The module must come from the core image library specified with DLBL and LIBDEF statements.

Stage 2 Security Exit

A user-supplied exit implements this control point. This point provides a standard interface for user ID and password verification and for establishing Connect:Direct functional authority, as well as file verification. While the exit can be used for many different purposes, the stage 2 security exit design provides the interface to your security system.

- ◆ The stage 2 security exit implements as an executable module.
- ◆ The module must come from the core image library specified with DLBL and LIBDEF statements.
- ◆ Activation of the stage 2 security exit is achieved by specifying SECURITY.EXIT=(modname.all) in the Connect:Direct initialization parameters.
- ◆ Sublibrary S of the source statement library NDM.SSLLIB contains a sample source program called DMCXSEC that can be used as a model.

Note: Since most VSE/ESA environments do not support a security system, an alternative to the stage 2 Security exit is the Connect:Direct authorization facility described in this chapter.

Run Job Security Exit

This control point provides a standard interface for security and syntactical verification of job streams before submitting them to the job entry system. The following requirements and restrictions apply:

- ◆ The Run Job exit is implemented as an executable module.
- ◆ The name of the module is user-definable but must not conflict with any Connect:Direct modules.
- ◆ You must activate the Run Job exit by specifying RUN.JOB.EXIT=(modname) in the Connect:Direct initialization parameters.
- ◆ The module must be link-edited, reentrant, and placed in a core image library that can be accessed by the NDM DTF.
- ◆ The source statement library, NDM.SSLIB, sublibrary A, contains a sample exit called DMCXRT.
- ◆ The sample exit can be used as a model to implement specific requirements.

Connect:Direct Authorization Facility

The Connect:Direct authorization facility can control access to Connect:Direct functions and serve as the source of security information as an alternative to the stage 1 signon exit and stage 2 security exit. If the Connect:Direct authorization facility is used, all Connect:Direct users must be identified in all nodes that will participate in Process execution.

Example of Authorization Facility Use

The following example shows how the Connect:Direct authorization facility can be used. In this example there are two Connect:Direct nodes, called SYSTEMA and SYSTEMB. Joe has access to SYSTEMA under the Connect:Direct userid of JOEA and access to SYSTEMB under the Connect:Direct userid of JOEB.

Joe requires two entries in the Connect:Direct authorization facility of each system as shown in the following figures. This gives him access to Connect:Direct on both systems and the authorization to move files between both systems.

SYSTEMA Authorization File			
Node	Connect:Direct User ID	Password	Authorized Functions
SYSTEMA	JOEA	[pswd]	Y, Y, N, Y,
SYSTEMB	JOEB	[pswd]	N, Y,

SYSTEMB Authorization File			
Node	Connect:Direct User ID	Password	Authorized Functions
SYSTEMA	JOEA	[pswd]	Y, Y, N, Y,
SYSTEMB	JOEB	[pswd]	N, Y,

The combination of logical node name and userid is used to access the Authorization File on the remote node to obtain the userid and password, as well as associated functional authority.

For example, if Joe were to send a file from SYSTEMA to SYSTEMB, the combination of SYSTEMA and JOEA would be used to access the authorization file on SYSTEMB. This entry would then be used to determine what Connect:Direct functional authority Joe has on SYSTEMB when coming from SYSTEMA.

Note: The Connect:Direct password is optional, but if specified in the Connect:Direct authorization facility, it must also be specified on the Connect:Direct SIGNON command and be available at Process execution time through the signon or SNODEID override.

See Chapter 4, *Maintaining User Authorization File* for information on how to maintain the user authorization file.

Using Optional Connect:Direct for VSE/ESA Exits

This chapter explains the installation steps for the following user exits:

- ◆ Statistics exit
- ◆ Submit exit
- ◆ Allocation exit
- ◆ I/O exit

Note: The High-Level Assembler for MVS & VM & VSE is required to assemble exits.

Statistics Exit

Connect:Direct generates and logs extensive statistics to an online journal. This information is recorded to the Connect:Direct statistics log as discrete records. These records are available in both formatted and unformatted form by means of the SELECT STATISTICS command. Each record contains information about a single event, and has an associated record type identifier which consists of two characters located at offset X'02' from the beginning of the record. For example, type CT is a copy termination record and FP is a flush process record.

Connect:Direct also provides a statistics exit that gives a user-written program access to the statistics records as they are generated. This exit can perform the following:

- ◆ Output the records, or data generated from the records, to a user-defined journal
- ◆ Preclude or allow the logging of any record by means of return codes to Connect:Direct

Warning: Statistics records are often useful or indispensable in debugging problems with Connect:Direct for VSE/ESA. Be aware that the exclusion of records from the statistics log can make problem determination by the Sterling Commerce Customer Services staff difficult or impossible in some cases.

The statistics exit runs as a subtask in the Connect:Direct DTF partition. Connect:Direct uses the STATISTICS.EXIT initialization parameter to specify the exit module name. The name is user-definable, but must not conflict with the name of any Connect:Direct module. If a user-defined journal is required, the necessary disk label (DLBL) statements must be added to the Connect:Direct start-up job stream. Connect:Direct for VSE/ESA provides the following sample statistics exits in the source library:

Statistics Exits	Description
STATEXIT	This sample exit simply checks for copy termination records. When Connect:Direct encounters a copy termination, the system issues a WTO.
STATEXMC	This sample exit precludes the logging of member copy (MC) records that have good return codes. It allows the logging of records of this type only when they have non-zero return codes. It also allows the logging of all other record types regardless of their return codes.

Calling Conventions of the Statistics Exit

Connect:Direct calls the statistics exit once for each statistics record that is generated. Standard linkage conventions apply.

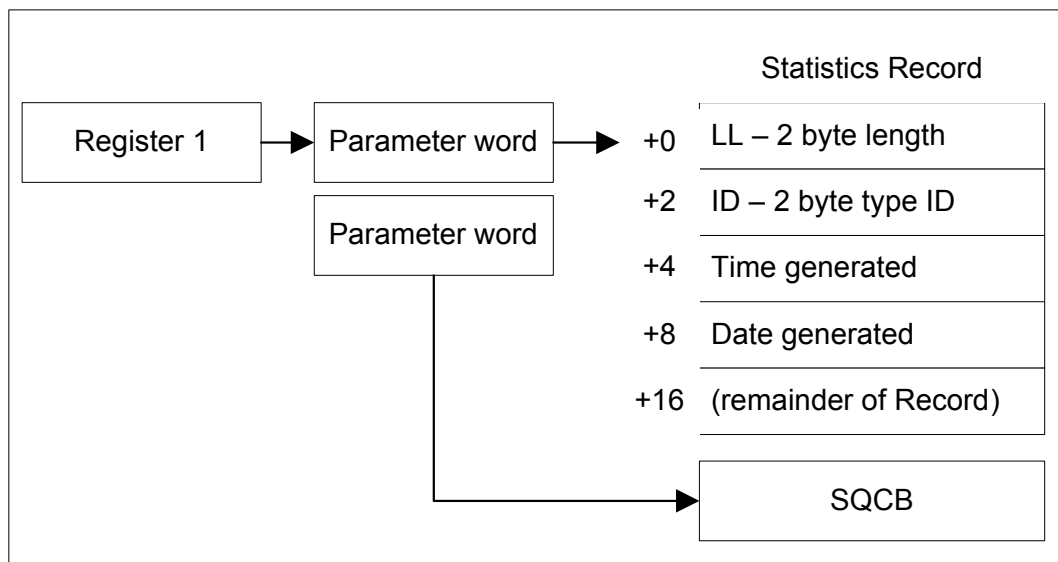
ESA Systems

As with the other Connect:Direct for VSE/ESA exits, on a VSE/ESA system, the system will load the exit above or below 16 megabytes depending on the exit RMODE attribute of the module. The system will also give the exit control in either 24-bit or 31-bit mode depending on its AMODE attribute. Unless the DTF initialization parameters specify ALLOC.STORAGE=BELOW, the exit should be linked with AMODE=31 and be coded to run in 31-bit mode. This is because the location of both the Connect:Direct provided save area and the exit parameter is above 16 megabytes.

The exit is given control with register 1 pointing at a list of two parameters. The first parameter is a pointer to the statistics record. The second parameter is a pointer to an SQCB, which is needed for setting a return code if record exclusion is desired. The first two bytes of the record contain the record length in binary. The third and fourth bytes of the record contain the two-character record identifier. The table on page 97 contains a list of the record type identifiers. The second word of the record contains the time of day that the record was generated, and the third word contains the date the record was generated. The remainder of the contents of the records varies depending on the specific record type. Assembler macros are provided in the source library to generate dummy sections (DSECTS) to map all the record types. The exit program should include the DSECTS that map whichever record types the exit will process.

The only output required from the exit by Connect:Direct is a return code set in the SQUSER field of the SQCB. This is used to indicate to Connect:Direct whether or not the record should be logged. A return code of 0 indicates that the record is to be logged. A return code of 4 indicates that the record should not be logged.

The figure below depicts the information passed to the exit.



Statistics Records

The statistics exit is called once for each statistics record generated in the DTF. The record can be of any of the various record types. The exit must examine the record type identifier at a displacement of X'02' bytes from the beginning of the record to determine the record type and the DSECT that describes its contents.

Connect:Direct for VSE/ESA Statistics Records

The following table lists the various statistics record types, their corresponding record type identifiers, and the name of the assembler macro in the source library that generates the DSECT describing the record contents.

Identifier	Description	Macro
PS	Process submit	DMPSSR
CT	Copy termination	DMCTR
MC	PDS member copy	DMFMCR
RT	Run Task	DMRTTR
RJ	Run Job	DMRJTR
SW	Submit within a Process	DMPSSR
WO	WTO	DMFWTOST
PT	Process termination	DMPTR

Identifier	Description	Macro
SD	Start Connect:Direct	DMSDCR
ST	Stop Connect:Direct	DMSTDRCR
SI	Signon	DMSFR
SO	Signoff	DMSFR
SP	Select Process	DMDTR
DT	Select Task	DMDTR
SS	Select Statistics	DMDTR
SN	Select Netmap	DMDTR
FP	Flush Process	DMFPTR
FS	Suspend Process	DMFPTR
CH	Change Process	DMCPTR
DP	Delete Process	DMDPTR
FT	Flush Task	DMDTR
TS	Suspend Task	DMFPTR
IU	Insert User	DMAER
SU	Select User	DMAER
DU	Delete User	DMAER
UU	Update User	DMAER
UM	Update Netmap	DMAER
XO	Trace on/off	DMXOR
SF	Statistics format	DMFSFREC
IF	Process modal - IF statement	DMMODAL
GO	Process modal - GOTO, ELSE, or EXIT statement	DMMODAL
NL	Process modal - EIF or PEND statement	DMMODAL
CS	Statistics command	DMFSCMDR
SC	Statistics control record	DMFSCR
S2	Statistics logging statistics	DMFS2R

Statistics Records

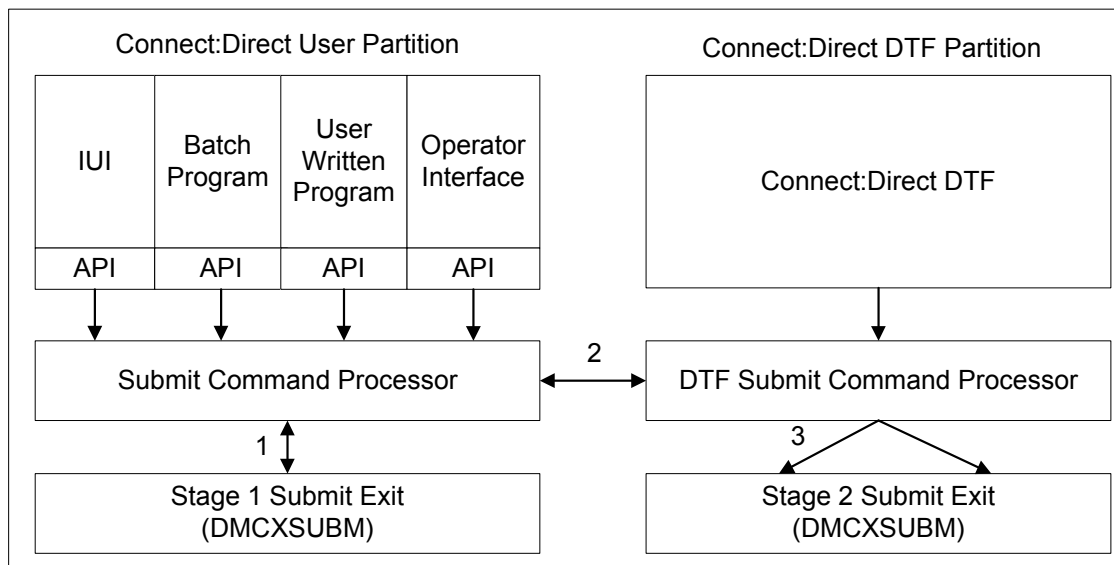
Control blocks for the statistics records shown in the following table are in the source statement library, type A. Information is passed to the statistics exit program through any of these control blocks. Users are responsible for determining how the control block information will be used within their programs.

Macro Name	Description
DMAER	Authorization event statistics record
DMCPTR	CHANGE PROCESS statistics record
DMCTR	COPY termination statistics record
DMDPTR	DELETE PROCESS statistics record
DMDTR	Display termination record
DMEVR	Event Services Command record
DMFMCR	Member Copy record
DMFPTR	FLUSH AND SUSPEND PROCESS statistics record
DMFS2R	Statistics Facility statistics record
DMFSCMD	Statistics file pair change record
DMFWTOST	WTO statistics record
DMGFTR	General function termination record
DMLSR	Log swap statistics record
DMMODAL	Process statement modal record
DMPIR	Process initiation record
DMPSSR	SUBMIT PROCESS statistics record
DMPTR	Process termination record
DMQCR	Queue change record
DMRJTR	RUN JOB termination record
DMRTTR	RUN TASK termination record
DMSDCR	Start CD command record
DMSFR	SIGNON statistics record
DMTPR	Throughput record
DMSTDCR	STOP NDM statistics record
DMSTEPR	Step initiation record
DMXOR	TRACE on/off statistics record

Submit Exit

The submit exit provides an interface to a user-written program when a Connect:Direct Process is submitted. With this interface, the user program can change Process information, such as Process name, priority, class, and secondary node, as well as copy step information such as data set name.

The following figure shows the execution order of the SUBMIT command.



Submit Exit Processing Flow

The processing flow for the submit exit is:

1. When a SUBMIT command or SUBMIT statement is issued, the API SUBMIT command processor calls the Stage 1 submit exit.
2. If the submit is successful, the API SUBMIT command processor calls the DTF SUBMIT command processor.
3. The DTF SUBMIT command processor calls the Stage 2 submit exit.

Note: If the SUBMIT command is executed from a Submit statement, the DTF Submit command processor first calls the Stage 1 submit exit and then the Stage 2 submit exit.

Implementation of Submit Exit

This section presents information about the following topics related to the submit exit:

- ◆ Interface specifications
- ◆ Control block format
- ◆ TCQ element (TCQE) control block (Process statement information)

- ◆ TCQ statement header (TCQSH)
- ◆ COPY control block
- ◆ RUN JOB control block
- ◆ Run Task control block
- ◆ Submit control block
- ◆ TCQE for If statements

Stage 1 Submit Exit

This control point executes in the user's address space when a user-written program contains a Submit command. This exit will not be called when a program contains a Submit statement.

- ◆ The Connect:Direct stage 1 submit exit is implemented as an executable module.
- ◆ The name of the module must be DMCXSUBM.
- ◆ The module must be reentrant.

Stage 2 Submit Exit

This control point executes in the DTF's partition when a user-written program contains a Submit command or a Submit statement.

- ◆ The stage 2 submit exit is implemented as an executable module.
- ◆ The name of the module is user-definable but must not conflict with any Connect:Direct load modules.
- ◆ Activation of the Stage 2 submit exit is achieved by specifying SUBMIT.EXIT= (modname) in the initialization parameters.
- ◆ The module must be link-edited, reentrant, and placed in a core image library that can be accessed by the DTF.

Control Block Format

Because the submit exits are invoked before the Process is actually submitted, some control block fields will not be filled in yet. This section presents control blocks used with all Process statements.

Upon entry into the user-written exit, Register 1 contains the address of the TCQE. A pointer to the TCQE Header (DMTCQE) is passed to the exit. Chained off the TCQE is the Command Statement Header (DMTCQSH) followed by the statement control block (Copy, Run Job, Run Task, or Submit).

If Process submission is rejected, a non-zero value is returned in Register 15. The exit should also set that value into the return code field (TQRTNCD) of the TCQE as well as setting a message ID in the TQMSGID field in the TCQE.

The following figure shows the layout of the TCQE. Note that DMxxxxxx represents the macro name for the statement (Copy, Run Job, Run Task, Submit, and so on) within the Process.

Displacement values found in the TCQE and the TCQSH are from the top of the TCQE. Displacement values found in the statement control blocks are from the top of the TCQSH associated with that statement control block.

L	L	TCQ HEADER	macro = DMTCQE dsect = TCQE
COMMAND STATEMENT HEADER SECTION			macro = DMTCQSH dsect = TCQSH
STATEMENT CONTROL BLOCK			macro = DMxxxxxx

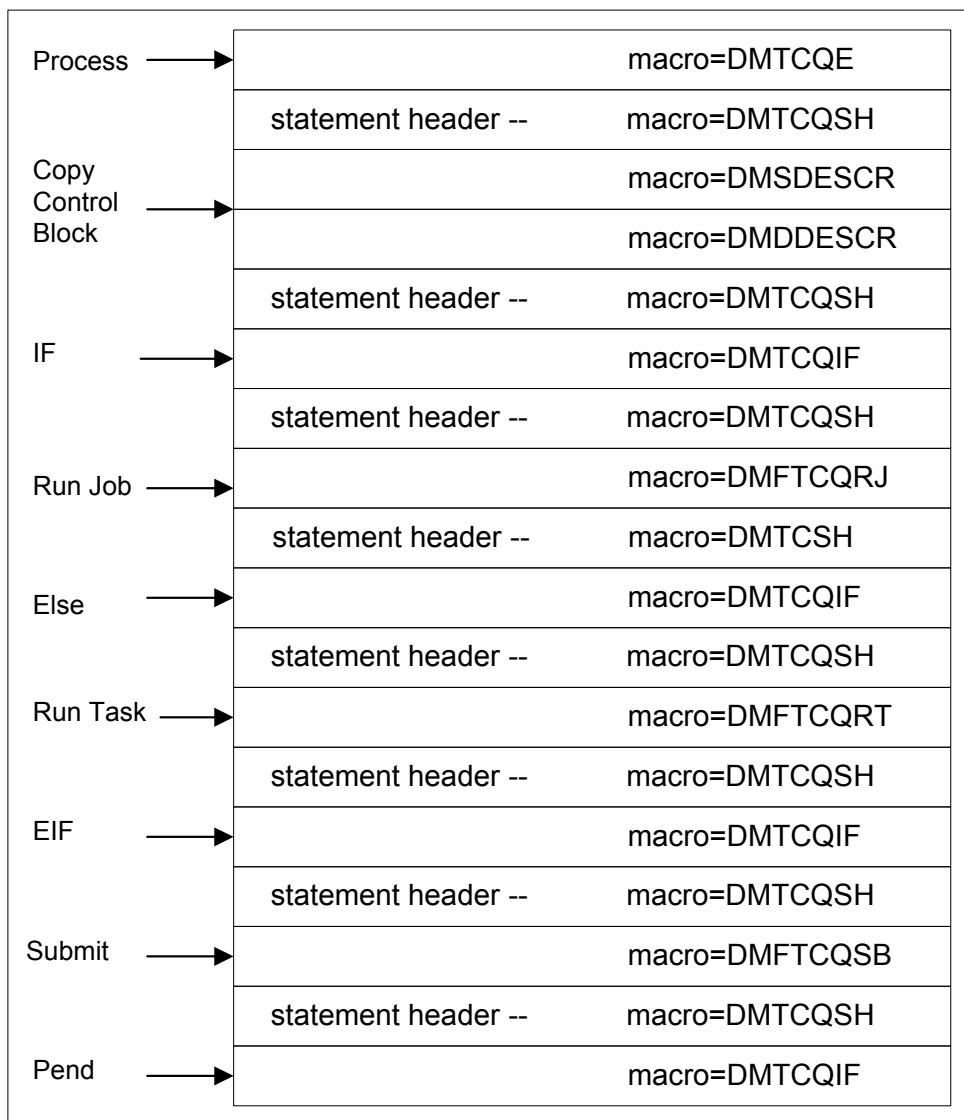
Example of Created Control Block

This example section shows how a Process is submitted and the control block that is created when the submit exit is invoked. The following Process called TEST01 is submitted.

```

TEST01  PROCESS  SNODE=THERE
STEP01  COPY   FROM (DSN=THIS.DATA.SET)   -
        TO     (DSN=THAT.DATA.SET DISP=OLD)
IF01    IF (STEP01=0) THEN
STEP02  RUN JOB   (DSN=mt (member) PNODE
ELSE
STEP03  RUN TASK (PGM=RTEXAMPL, PARM=(CL44 'THIS.DATA.SET' )
        PNODE
EIF
STEP04  SUBMIT  DSN=mt (member) HOLD=Y
    
```

The following figure shows the resulting layout of the Process control block after submission of the Process named TEST01.



Modifiable TCQE Fields

The following table describes TCQE fields that can be examined or modified by the submit exit.

TCQE Field	Contains
TQCBHLNG	length of the entire TCQE. This length added to the address of the TCQE gives the address of the TCQSH.
TQSTMTN	number of statements in this Process.
TQUNODE	symbolic node name for the submitter of this Process.

TCQE Field	Contains
TQUID	user ID for the submitter of this Process.
TQUPAS	password for the submitter of this Process.
TQPUID	security user ID at the primary node.
TQOPPAS	old security password at the primary node.
TQNPPAS	new security password at the primary node.
TQSUID	security user ID at the secondary node.
TQOSPAS	old security password at the secondary node.
TQNSPAS	new security password at the secondary node.
TQRTNCD	Process completion code. The user exit should change this when an error is encountered in the exit or if the Process is no longer to be submitted upon return from the exit.
TQMSGID	Process message ID. The user exit should enter a message ID related to any return codes set in the exit.
TQCSPRD	displacement to the first Process statement from the TCQE. This length added to the address of the TCQE gives the address of the TCQSH.
TQPRSBYT	parallel session class. See <i>Conversion of Parallel Session Values</i> on page 105 for details.
TQPRSBIT	parallel session class. See <i>Conversion of Parallel Session Values</i> on page 105 for details.
TQPROCNM	name of the Process being submitted.
TQSCHDTE	Julian date the Process is scheduled to be submitted.
TQSCHTME	time of day the Process is scheduled to be submitted.
TQSCHDAY	day of the week that the Process is scheduled to be submitted.
TQPRTY	priority for Process selection.
TQRETAIN	retain status for the Process.
TQSELDTE	Julian date a retained Process is to be submitted.
TQTODFLG	contains the following interval control flags: <ul style="list-style-type: none"> ◆ If TQTODTD is on, a Process has a scheduled time and date it is to be submitted ◆ If TQTOTME is on, a Process has a scheduled time it is to be submitted ◆ If TQTODDAY is on, a Process has a scheduled day of the week it is to be submitted ◆ If TQTODINT is on, a Process is scheduled to run when a specified interval expires
TQPNODE	symbolic node ID of the primary node.
TQSNODE	symbolic node ID of the secondary node.

TCQE Field	Contains
TQSTATUS	Process status.

Conversion of Parallel Session Values

The session class value is stored in two bytes in the TCQE (TQPRSBYT and TQPRSBIT). The class that was specified can be derived from these values. The following table shows a sample of the two bytes for the first 16 classes (maximum can be 256).

TQPRSBYT	TQPRSBIT	CLASS
00	80	1
00	40	2
00	20	3
00	10	4
00	08	5
00	04	6
00	02	7
00	01	8
01	80	9
01	40	10
01	20	11
01	10	12
01	08	13
01	04	14
01	02	15
01	01	16

Allocation Exit

The Connect:Direct allocation exit provides an interface to a user-written program. If you supply a user exit in the initialization parameters, Connect:Direct will invoke the exit prior to any allocation activity by the receiving Connect:Direct.

Through the exit you can change information that Connect:Direct uses during the allocation process. You can examine or modify information such as data set name (DSN) and type record name or set fields to terminate the copy step prior to allocation.

Connect:Direct provides a sample allocation exit, DMGALOEX, located in the source library. DMGALOEX demonstrates how to access the VSAMPL and the TCQE, and both the source and destination description in the TCQE. It shows how to change a value in the Data Set Description Control Blocks (DMDDESCR or DMSDESCR) and set a return code and message ID before return.

Observe the following restrictions and requirements:

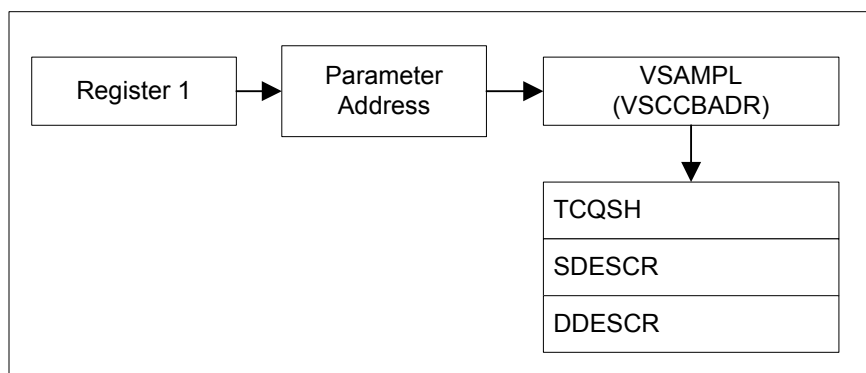
- ◆ The name of the allocation exit load module is user-definable, but it must not conflict with the name of any other Connect:Direct load modules.
- ◆ To activate the exit, specify ALLOCATION.EXIT=modname in the Connect:Direct initialization parameter file. The allocation exit module must be link-edited as reentrant and placed in a load library that can be accessed by the Connect:Direct DTF.

How the Allocation Exit Executes

The allocation exit executes in the DTF partition when the following conditions exist:

- ◆ The allocation exit is specified in the initialization parameters.
- ◆ A file is being received, and the PROCESS step that has initiated the Copy is not in restart mode.

The following figure shows how the parameter list for the allocation exit is structured.



The following table is a list of the allocation exit parameters:

Parameter	Explanation
R1	Register 1—contains the address of a standard parameter list upon entry into the user-written allocation exit
PLIST	Stands for standard parameter list
VSAMPL	Stands for VSAM parameter list, whose address is the first fullword in the PLIST
VSCCBADR	Address of the Process step header and is contained in VSAMPL
TCQSH	Process step header portion of the Copy control block (each step of a process generates a TCQSH)
SDESCR	Source data set-descriptor portion of the Copy control block
DDESCR	Destination data-set descriptor portion of the Copy Control Block (a sample of the DDESCR Control Block is included in this section)

The macros listed in the following table are used to map the control block structures described in the previous table:

Macro	Explanation
DMVSAMPL	Macro that defines the VSAMPL control block
DMTCQSH	Macro that defines the TCQSH portion of the Copy control block
DMSDESCR	Macro that defines the SDESCR portion of the Copy control block
DMDDESCR	Macro that defines the DDESCR portion of the Copy control block

Calculating Addresses and Values

Upon entry into the user-written allocation exit, Register 1 contains the address of a standard parameter list. The first and only entry in the PLIST contains the address of the VSAMPL. The VSCCBADR field in the VSAMPL contains the address of the Process step header, TCQSH. The SDESCR and DDESCR following the TCQSH can be found by adding displacements to the TCQSH address.

- ◆ To calculate the location of SDESCR, add the length of TCQSH (TSHCBHLN) to the TCQSH address (VSCCBADR).
- ◆ To calculate the location of DDESCR, add the length of SDESCR (S1SVSLNG) to the SDESCR address.

All displacement values in the Copy control block are referenced from the beginning of the TCQSH control block.

SDESCR and DDESCR contain both fixed length fields and offsets to variable length fields. Any fixed length field in DDESCR can be modified by the allocation exit.

The fields that are referenced within SDESCR and DDESCR using displacement values are variable in length and should **not** be modified by the allocation exit.

The DSN field is created with enough space to hold a 100-character name, only if the TSEXPDSN bit is on (set) in the TCQSH. If the TSEXPDSN bit is off in the TCQSH, then the Copy control block does not contain the room to expand the DSN. This means that this copy originated from a node that did not build the Copy control block with an expandable DSN field.

The DSN field can be found by adding D1DDSN to the address of the TCQSH. The DSN field contains a two byte length field followed by the DSN. Even though the field can be up to 100 bytes long, the two byte length field contains the actual length of the DSN. If the DSN is changed, the two byte length field must be modified accordingly. The other variable length fields are created with their current values and *cannot* be lengthened. Do not modify the D1DDSN field.

When allocating the destination file, the Connect:Direct software first uses values from DDESCR, as specified in the Connect:Direct Copy statement. Any values needed but not set in DDESCR are taken from the Type record, if one was specified. Any remaining values are taken from the SDESCR portion of the Copy control block.

Note: If the D1DTYPE field is modified by the allocation exit, the exit must clear any fields in the DDESCR portion of the Copy control block that would override the corresponding Type fields from the Type record.

Copy Control Block Definitions

The following Copy control block definitions can be generated within the allocation exit program by including the macro name followed by DSECT=YES, as follows:

DMVSAMPL	DSECT=YES
DMTCQSH	DSECT=YES
DMDDESCR	DSECT=YES
DMSDESCR	DSECT=YES

Copy Control Block Modifications

Only the VSAMPL and DDESCR control blocks should be modified by the user. For the VSAMPL control block, the user is required to modify the VSRTNCD and VSMSGID fields. The following describes how to make these modifications:

- ◆ The VSRTNCD field (and R15) must be set to zero (0) to allow the Copy step to execute.
- ◆ The VSRTNCD field (and R15) must be set to a non-zero value to fail the Copy step.
- ◆ The user must insert a message ID into the VSMSGID field if the VSRTNCD is set to a non-zero value. Precaution should be taken to not duplicate existing message IDs.
- ◆ The user should also place message text corresponding to these message IDs in the message file.

The next section describes the DDESCR control block format and how to modify it.

DDESCR Control Block Format

The following example shows the DDESCR control block format. Information underlined within the control block is modifiable. The modifiable fields are described immediately following this figure.

Note: Turning a flag on means setting the flag bit in the byte to **1**. Turning a flag off means setting the flag bit in the byte to **0**.

```

*****
*
*           DMDESCR VERSION 1 RELEASE 5 MODIFICATION 0           1991 *
*   TYPE: DATA SET CONSTANT                                     *
*   NAME: DESTINATION DATA SET DESCRIPTION                     *
*
*-----*
DDESCR      DS      OF          /*DESTINATION DATA SET DESCRIPTION */
*-----*
*   TYPE: DATA SET DSECT                                     *
*-----*
D1DVSLNG    DC      Y(DDESCR#) /*LENGTH OF DEST. DATA SET VARIABLE */
D1DDESCR    DC      CL8'DDESCR /* VARIABLE SECTION NAME "DDESCR" */
BOOLEAN     D1FLAG1, (D1FIL1, D1FIL2, D1FIL3, D1TYPE, D1MEMNAM,
                   D1QUALFR, D1DSN, D1PWD)
*D1FLAG1    DC      XL1'00-    /*FLAGS FOR PARM PRESENT          */
*D1FIL1     EQU     X'80'      /*FILTER 1 SPECIFIED      */
*UNUSED     EQ      X'01'      /*UNUSED                   */
*D1FIL2     EQU     X'40'      /*FILTER 2 SPECIFIED      */
*D1FIL3     EQU     X'20'      /*FILTER 3 SPECIFIED      */
*D1TYPE     EQU     X'10'      /*DATA TYPE SPECIFIED     */
*D1MEMNAM    EQU     X'08'      /*MEMBER NAME SPECIFIED   */
*D1QUALFR   EQU     X'04'      /*DATA SET QUALIFIER SPECIFIED */
*D1DSN      EQU     X'02'      /*DATA SET NAME SPECIFIED  */
*D1PWD      EQU     X'01'      /*DATA SET NAME PASSWORD  */
BOOLEAN     D1FLAG, (D1BLKSZE, D1BUFNUM, D1DENSTY, D1DSORGN,
                   D1KEYLN, D1LIMCNT, D1LRECLN, D1OPTCDE)
*D1FLAG2    DC      XL1'00'    /*FLAG FOR DCB PARMS     */
*D1BLKSZE   EQ      X'80'      /*BLOCK SIZE SPECIFIED    */
*D1BUFNUM   EQ      X'40'      /*NUMBER OF BUFFERS SPECIFIED */
*D1DENSTY   EQ      X'20'      /*DENSITY SPECIFIED       */
*D1DSORGN   EQ      X'10'      /*DATA SET ORGANIZATION   */
*D1KEYLN    EQ      X'08'      /*KEYLENGTH SPECIFIED     */
*D1LIMCNT   EQ      X'04'      /*LIMCT SPECIFIED        */
*D1LRECLN   EQ      X'02'      /*LOGICAL RECORD LENGTH SPECIFIED */
*D1OPTCDE   EQ      X'01'      /*OPTION CODE SPECIFIED   */
BOOLEAN     D1FLAG3, (D1RECFMT, D1RKYP, D1TRKCH, D1CYLOFL, D1NTM)
*D1FLAG3    DC      XL1'00'    /*FLAG FOR DCB PARM     */
*D1RECFMT   EQ      X'80'      /*RECORD FORMAT SPECIFIED */
*D1RKYP     EQ      X'40'      /*RKP SPECIFIED          */
*D1TRKCH    EQ      X'20'      /*TRTCH SPECIFIED       */
*D1CYLOFL   EQ      X'10'      /*CYL OVERFLOW SPECIFIED */
*D1NTM      EQ      X'08'      /*NUMBER OF INDEXED TRACKS SPECIFIED*/
*UNUSED     EQ      X'04'      /*UNUSED                 */
*UNUSED     EQ      X'02'      /*UNUSED                 */
*UNUSED     EQ      X'01'      /*UNUSED                 */
BOOLEAN     D1FLAG4, (D1DS, D1LABEL, D1PSSD, D1NOPWRD, D1RETPRD,
                   D1EXPDTE)
*D1FLAG4    DC      XL1'      00' /*FLAG FOR LABEL PARM    */
*D1DS       EQ      X'80'      /*DATA SET SEQUENCE NUMBER SPECIFIED */
*D1LABEL    EQ      X'40'      /*LABEL TYPE SPECIFIED   */
*D1PSSWD    EQ      X'20'      /*PASSWORD SPECIFIED     */
*D1NOPWRD   EQ      X'10'      /*NO PASSWORD SPECIFIED  */
*D1RETPRD   EQ      X'08'      /*RETENTION PERIOD SPECIFIED */
*D1EXPDTE   EQ      X'04'      /*EXPIRATION DATE SPECIFIED */
*UNUSED     EQ      X'02'      /*UNUSED                 */
*UNUSED     EQ      X'01'      /*UNUSED                 */

```

(continued)

BOOLEAN	D1FLAG5, (D1BLK, D1TRK, D1CYL, D1REC)			
*D1BLK	EQ	X'80'	/*ALLOCATION IN BLOCKS SPECIFIED	*/
*D1TRK	EQ	X'40'	/*ALLOCATION IN TRACKS SPECIFIED	*/
*D1CYL	EQ	X'20'	/*ALLOCATION IN CYLINDERS SPECIFIED	*/
D1REC	EQ	X'10	/*ALLOCATION IN RECORDS SPECIFIED	*/
*UNUSED	EQ	X'08'	/*UNUSED	*/
*UNUSED	EQ	X'04'	/*UNUSED	*/
*UNUSED	EQ	X'02'	/*UNUSED	*/
*UNUSED	EQ	X'01'	/*UNUSED	*/
BOOLEAN	D1FLAG6, (D1SECALL, D1DRBLK, D1RELEAS, D1CONTIG, D1MXIG, D1ALX, D1ROUND)			-
*D1FLAG6	DC	XL1'00'	/*FLAG FOR SPACE PARM	*/
*D1SECALL	EQ	X'80'	/*SECONDARY ALLOCATION SPECIFIED	*/
*D1DRBLK	EQ	X'40'	/*DIRECTORY BLOCKS SPECIFIED	*/
*D1RELEAS	EQ	X'20'	/*RELEASE SPECIFIED	*/
*D1CONTIG	EQ	X'10'	/*CONTIG SPECIFIED	*/
*D1MXIG	EQ	X'08'	/*MXIG SPECIFIED	*/
*D1ALX	EQ	X'04'	/*ALX SPECIFIED	*/
*D1ROUND	EQ	X'02'	/*ROUND SPECIFIED	*/
*UNUSED	EQ	X'01'	/*UNUSED	*/
BOOLEAN	D1FLAG7, (D1GRPTYP, D1UNTCNT, D1P, D1DEFER)			
*D1FLAG7	DC	XL1'00'	/*FLAG FOR UNIT PARM	*/
*D1GRPTYP	EQ	X'80'	/*TYPE OR GROUP NAME SPECIFIED	*/
*D1UNTCNT	EQ	X'40'	/*UNIT COUNT IS SPECIFIED	*/
*D1P	EQ	X'20'	/*P SPECIFIED	*/
*D1DEFER	EQ	X'10'	/*DEFER SPECIFIED	*/
*UNUSED	EQ	X'08'	/*UNUSED	*/
*UNUSED	EQ	X'04'	/*UNUSED	*/
*UNUSED	EQ	X'02'	/*UNUSED	*/
*UNUSED	EQ	X'01'	/*UNUSED	*/
BOOLEAN	D1FLAG8, D1VOLSER, D1PRIV, D1RETAIN, D1VOLSEQ, D1VOLCNT, D1VOLRF)			-
*D1FLAG8	DC	XL1'00'	/*FLAG FOR VOLUME PARM	*/
*D1VOLSER	EQ	X'80'	/*VOL SER SPECIFIED	*/
*D1PRIV	EQ	X'40'	/*PRIVATE SPECIFIED	*/
*D1RETAIN	EQ	X'20'	/*RETAIN SPECIFIED	*/
*D1VOLSEQ	EQ	X'10'	/*VOLUME SEQUENCE NUMBER SPECIFIED	*/
*D1VOLCNT	EQ	X'08'	/*VOLUME COUNT SPECIFIED	*/
*D1VOLRF	EQ	X'04'	/*VOLUME REFERBACK	*/
*UNUSED	EQ	X'02'	/*UNUSED	*/
*UNUSED	EQ	X'01'	/*UNUSED	*/
BOOLEAN	D1FLAG9, (D1SOCL, D1SOPN, D1SOCN)			
*D1FLAG9	DC	XL1'00'	/*FLAG FOR SYSOUT PARM	*/
*D1SOCL	EQ	X'80'	/*SYSOUT CLASS SPECIFIED	*/
*D1SOPN	EQ	X'40'	/*SYSOUT PROGRAM NAME SPECIFIED	*/
*D1SOCN	EQ	X'20'	/*SYSOUT CODE NAME SPECIFIED	*/
*UNUSED	EQ	X'10'	/*UNUSED	*/
*UNUSED	EQ	X'08'	/*UNUSED	*/
*UNUSED	EQ	X'04'	/*UNUSED	*/
*UNUSED	EQ	X'02'	/*UNUSED	*/
*UNUSED	EQ	X'01'	/*UNUSED	*/

(continued)

```

BOOLEAN  D1FLAG10, (D1DEST, D1COP, D1DCBREF, D1MVSVGP, D1LINK,
                  D1VSMCAT, D1SYSOP, D1PROTCT)
* D1FLAG10  DC      XL1'00'  /*FLAG FOR MISC PARM          */
* D1PDSSEL  EQ      X'04'   /*SELECT LIST SPECIFIED          */
* D1DEST    EQ      X'80'   /*DESTINATION OF OUTPUT WAS SPECIFIED */
* D1COP     EQ      X'40'   /*NUMBER OF SYSOUT COPIES WAS SPECIFIED */
* D1DCBREF  EQ      X'20'   /*DCB=DSNAME                     */
* D1BMSVGP  EQ      X'10'   /*MSVGP=XXXXXXXXX                */
* D1LINK    EQ      X'08'   /*'LINK=' KEYWORD WAS SPECIFIED    */
* D1VSMCAT  EQ      X'04'   /*'VSAMCAT=' KEYWORD WAS SPECIFIED  */
* D1SYSOP   EQ      X'02'   /*'SYSOPTS=' KEYWORD WAS SPECIFIED  */
* D1PROTCT  EQ      X'01'   /*'PROTECT=YES' WAS SPECIFIED      */
BOOLEAN  D1FLAG11, (D1COMP, D1YCKPT, D1DOCKPT, D1RSTRT, D1NOCKPT,
                  D1IODRVR, D1CKPTPO, D1NGCKPT)
***
*** PLEASE NOTE: ONLY THE D1DOCKPT AND D1IODRVR BITS HAVE MEANING
*** THE OTHER BITS ARE MIRRORS OF THE SDESCR SECTION.
*** THE D1DOCKPT BIT IS SET ON BY THE RECEIVING SIDE
*** TO INDICATE CHECKPOINTING IS TO BE DONE. IF BOTH
*** THE D1IODRVR AND D1DOCKPT BITS ARE ON, C:D WILL DO
*** THE FOLLOWING ON THE SENDING SIDE...
*** 1. IF SENDING SIDE IS ALSO I/O DRIVER THEN S1DOCKPT MUST HAVE
*** BEEN TURNED ON BY INPUT USER I/O DRIVER...ELSE ERROR
*** 2. IF SENDING SIDE IS NOT A USER I/O DRIVER THEN INPUT SIDE MUST
*** BE DSORG=PS (IF D1CKPTPO IS OFF)OR DSORG=PO(IF D1CKPTPO IS ON)
*** ...ELSE ERROR
***
* D1FLAG11  DC      XL1'00' /*FLAG FOR COMPRESSION          */
* D1COMP    DC      X'80'   /*COMPRESSION SPECIFIED          */
* D1YCKPT   DC      X'40'   /*CHECKPOINTING EXPLICITLY SPECIFIED */
* D1DOCKPT  DC      X'20'   /*CHECKPOINTING REQUESTED BY RECEIVER */
* D1RSTRT   DC      X'10'   /*PROCESS NOT EXEC THE FIRST TIME    */
* D1NOCKPT  DC      X'08'   /*CKPT NEVER TAKEN                */
* D1IODRVR  DC      X'04'   /*RECEIVER IS A USER I/O DRIVER    */
* D1CKPTPO  DC      X'02'   /*I/O DIRVER SAYS DO PDS CHECKPOINTING */
* D1NGCKPT  DC      X'01'   /*SRC SIDE CKPTING REQUESTED        */
BOOLEAN  D1FLAG12, (D1PDSNOD, D1PDSNGL, D1PDSNOA, D1PDSNOR, D1PDSEXL,
                  D1PDSSEL, D1NOBLOK, D1SPOOL)
* D1FLAG12  DC      XL1'00' /*FLAG FOR PDS SUPPORT          */
* D1PDSNOD  EQ      X'80'   /*PDS.DIR = NO SPECIFIED          */
* D1PDSNGL  EQ      X'40'   /*SINGLE MEMBER SPECIFIED          */
* D1PDSNOA  EQ      X'20'   /*ALIAS=NO SPECIFIED              */
* D1PDSNOR  EQ      X'10'   /*REPLACE = NO SPECIFIED          */
* D1PDSEXL  EQ      X'08'   /*EXCLUDE LIST SPECIFIED          */
* D1NOBLOK  EQ      X'02'   /*CANNOT SUPPORT BLOCK MODE        */
* D1SPOOL   EQ      X'01'   /*SPOOL DATASET                   */
DC      CL1' ' /*RESERVED FOR COMPRESSION CHARACTER */
DC      X'00'  /*ALIGNMENT                        */
DC      F'00'  /*RESERVED FOR CHECKPT INTERVAL    */
DC      F'00'  /*ALIGNMENT                        */
DC      D'00'  /*RESERVED - # OF BYTES RECEIVED    */
D1EXIT1    DC      CL8' ' /*DATA FILTER EXIT 1               */
D1EXIT2    DC      CL8' ' /*RESERVED                          */
D1EXIT3    DC      CL8' ' /*RESERVED                          */
D1DTYPE    DC      CL8' ' /*DATA TYPE CONNECT:Direct DEFINED */
*          /*C'BINARY',C'TEXT',              */
*          /*C'DF'                          */

```


(continued)

```

D1DDSTYP      DC      CL4' ' /*THE DESTINATION DATA SET TYPE      */
*              /*(ESDS, KSDS, RRDS, PDS, SAM, LIB)  */
D1DDDISP1     DC      C' ' /*SOURCEFILE DISPOSITION STATUS "N"=NEW*/
*              /*"O"=OLD "S" = SHR "M"= MOD "R" = RPL */
D1DDDISP2     DC      C' ' /*SOURCE FILE DISPOSITION WITH NORMAL */
*              /*TERMINATION"D" = DELETE , "K" = KEEP */
*              /*"C" = CATALOG                      */
D1DDDISP3     DC      C' ' /*SOURCE FILE DISPOSITION WITH ABNORMAL*/
*              /*TERMINATION "D"=DELETE , "K"=KEEP  */
*              /*"C" = CATALOG                      */
              DC      XL1'00' /*ALIGNMENT                          */
D1BLKSIZ      DC      CL5' ' /*BLOCK SIZE                          */
D1BUFNO       DC      CL2' ' /*NUMBER OF I/O BUFFERS              */
D1DEN         DC      CL1' ' /*TAPE DENSITY                       */
D1DSORG       DC      CL4' ' /*DATA SET ORGANIZATION              */
D1KEYLEN      DC      CL3' ' /*KEY LENGTH                          */
D1LIMCT       DC      CL3' ' /*LIMIT FOR EXTENDED SEARCH          */
D1LRECL       DC      CL5' ' /*LOGICAL RECORD LENGTH              */
D1OPTCD       DC      CL6' ' /*OPTIONAL SERVICE NUMBER            */
D1RECFM       DC      CL4' ' /*RECORD FORMAT                      */
D1RKP         DC      CL5' ' /*RELATIVE KEY POSITION                */
D1TRTCH       DC      CL2' ' /*7- TRACK RECORDING MODE            */
D1CYLO        DC      CL2' ' /*TRKS IN CYL OVERFLOW               */
D1NTMIN       DC      CL2' ' /*NUMBER OF TRKS FOR CYL INDEX       */
D1DSSEQ       DC      CL5' ' /*DATA SET SEQUENCE NUMBER           */
D1LABTYP      DC      CL3' ' /*LABEL TYPE                          */
D1RETPD       DC      CL4' ' /*RETENTION PERIOD                   */
D1EXPDT       DC      CL5' ' /*EXPIRATION DATE YYDDD              */
D1LOCTYP      DC      CL3' ' /*ALLOCATION TYPE - CYL,TRK,BLK,REC   */
D1PRILOC      DC      CL8' ' /*PRIMARY ALLOCATION AMOUNT           */
D1SECLOC      DC      CL8' ' /*SECONDARY ALLOCATION AMOUNT         */
D1DIRBLK      DC      CL5' ' /*NUMBER OF DIRECTORY BLOCKS         */
D1AVGBLK      DC      CL5' ' /*AVERAGE BLOCK SIZE FOR SPACE PARM  */
D1UNIT        DC      CL8' ' /*UNIT TYPE OR GROUP NAME            */
D1UNITCT      DC      CL2' ' /*UNIT COUNT                          */
D1VOLSEQ      DC      CL4' ' /*VOLUME SEQUENCE NUMBER             */
D1VOLCT       DC      CL4' ' /*VOLUME COUNT                       */
D1SYSCL       DC      CL1' ' /*SYSOUT CLASS                       */
D1SYSPN       DC      CL8' ' /*SYSOUT PROGRAM NAME                */
D1SYSCN       DC      CL4' ' /*SYSOUT CODE NAME                   */
D1DESTN       DC      CL8' ' /*DESTINATION OF OUTPUT               */
D1COPY        DC      CL3' ' /*NUMBER OF SYSOUT COPIES            */
D1PASSWD      DC      CL8' ' /*DATA SET PASSWORD                  */
              DC      F'00' /*RESERVED                            */
D1EXDISP      DC      H'00' /*DISPLACEMENT TO CCB EXPANSION SECT.*/
              DC      H'00' /*RESERVED..USED IN SDESCR FOR CKPT PTR*/
              DC      CL5' ' /*RESERVED FOR AUTH FILE LRECL       */
              DC      X'00' /*ALIGNMENT                          */
D1DMEMB       DC      H'00' /*DISPLACEMENT TO MEMBER NAME FIELD  */
D1DQUAL       DC      H'00' /*DISPLACEMENT TO QUALIFIER FIELD     */
D1DDSN        DC      H'00' /*DISPLACEMENT TO DATA SET NAME FIELD*/
D1DVOLN       DC      H'00' /*DISPLACEMENT TO VOLUME SERIAL NUMBER*/
D1DCBDSN      DC      H'00' /*DISPLACEMENT TO DCB REFERENCE NAME  */
D1VOLREF      DC      H'00' /*DISPLACEMENT TO THE VOLUME REFERENCE*/
D1DMEBEX      DC      H'00' /*DISPLACEMENT TO THE EXCLUDE LIST    */
              DS      0F /*CONTROL BLKS MUST BE FULLWORD ALIGNED*/
DDESCR#       EQU      *-DDESCR /* SIZE OF THE DDESCR FIXED SECTION  */
*****

```

Note: Do not modify displacement fields.

DDESCR Modifiable Fields

The following table lists the modifiable fields in the DDESCR portion of the Copy control block:

Field	Explanation
D1DTYPE	Entry in the Connect:Direct type defaults file
D1BLKSIZ	Block size To use the block size indicated in the destination data set description portion of the copy control block, set the D1BLKSZE flag on. To use the block size indicated in the Type record, set the D1BLKSZE flag off.
D1DEN	Tape density To use the tape density indicated in the destination data set portion of the Copy control block, set the D1DENSTY flag on.
D1DSORG	Data set organization To use this value, set the D1DSORGN flag on. To use the value indicated in the type record, set the D1DSORGN flag off.
D1LRECL	Logical record length To use this value, set the D1LRECLN flag on. To use the value indicated in the type record, set the D1LRECLN flag off.
D1RECFM	Record format To use this value, set the D1RECFMT flag on. To use the value indicated in the type record, set the D1RECFMT flag off.
D1RKP	Relative key position To use this value, set the D1RKYP flag on.
D1TRTCH	7-track recording mode To use this value, set the D1TRKTCH flag on.
D1LABTYP	Label type To use this value, set the D1LABEL flag on.
D1RETPD	Retention period To use this value, set the D1RETPRD flag on. To use the value indicated in the D1EXPDT field, set the D1RETPRD flag off.
D1EXPDT	Retention period To use this value, set the D1EXPDTE flag on.
D1PRILOC	Primary allocation amount All of the following bits must be set off to use the space allocation values specified in the type record: D1TRK D1CYL D1BLK.

Field	Explanation
D1SECLOC	<p>Secondary allocation amount</p> <p>To use this value, set the D1SECALL flag on.</p> <p>All of the following bits must be set off to use the space allocation values specified in the type record:</p> <p>D1TRK, D1CYL, D1BLK.</p>
D1DIRBLK	<p>Number of directory blocks</p> <p>To use this value, set the D1DRBLK flag on.</p> <p>All of the following bits must be set off to use the space allocation values specified in the type record:</p> <p>D1TRK, D1CYL, D1BLK.</p>
D1UNIT	<p>Unit type or group name</p> <p>To use this value, set the D1GRPTYP flag on. To use the value indicated in the type record, set the D1GRPTYP off. All of the following bits must be set off to use the UNIT value specified in the type record:</p> <p>D1UNCNT, D1P, D1DEFER, D1GRPTYP.</p>
D1VOLSEQ	<p>Volume sequence number</p> <p>To use this value, set the D1VOLSQ flag on.</p>
D1VOLCT	<p>Volume count</p> <p>To use this value, set the D1VOLCNT flag on.</p>
D1PASSWD	<p>Data set password</p> <p>To use this value, set the D1PWD flag on.</p>
D1DMEMB	Displacement to the member name field*
D1DDSN	Displacement to the data set name field*
D1DVOLN	Displacement to the volume serial number*

*You must not modify the fields D1DMEMB, D1DDSN, and D1DVOLN. These fields represent displacements to their corresponding values. However, after calculating the address of the values (by adding the displacement to the address of the TCQSH) the actual values can be changed by Connect:Direct. At the calculated address, you will find a halfword field representing the length of the data that follows. If the length of the data changes, this halfword must also be changed to reflect the new length. If the displacement to one of these fields is 0, do not insert a value or displacement. For example, if D1DMEMB=0, no member name was specified and a member name cannot be inserted. Do not set on the D1MEMNAM flag if D1DMEMB=0.

The same applies to D1DDSN. For D1DVOLN, if the displacement is 0, you cannot modify this field or turn on the D1DVOLSER flag. Also, you cannot add volume serial numbers to this list. You can delete volume serial numbers from the list or change the volume serial number. If volume serial numbers are deleted, decrement the length field by 6 for each one deleted. If all volume serial numbers are deleted, make D1DVOLN=0 and turn off the D1VOLSER bit.

I/O Exit

The Connect:Direct I/O Exit provides an interface to user-written programs, allowing them to read and write data to or from a file whose organization Connect:Direct does not support or would improperly access. Examples of this are direct access (DA) data sets with track reference information, and power spool files.

Implementing the I/O Exit

The following items should be taken into consideration if you plan to use I/O Exit.

- ◆ All I/O Exits must be re-entrant, follow IBM Assembler linkage standards and reside as a link edited phase accessible by the node where they will be referenced. These exits must not alter any Control Blocks (except in the EXTTCB as indicated). If other Control Block fields are altered, the results are unpredictable.
- ◆ If an ALLOCATION EXIT is specified, it will not be given control when the COPY statement contains an IOEXIT keyword.
- ◆ Any message IDs specified by an I/O Exit should be added to the Message file. See Chapter 8, *Maintaining Connect:Direct for VSE/ESA* for instructions to add messages to the message file.

A sample I/O Exit is provided in the source library, type S (member name NDMIOX01); however, I/O Exits will need to be written to satisfy your specific data set requirements.

After a user defined I/O Exit is written, it can be implemented by specifying the exit name on the IOEXIT keyword on a COPY statement, or by specifying the exit name on the IOEXIT keyword on the INSERT and UPDATE TYPE file commands.

Specifying the I/O Exit on the COPY Statement

One method of indicating that an I/O Exit will be used is to include the IOEXIT keyword on the COPY statement. The following is an example of the IOEXIT format on the COPY statement:

```
IOEXIT=exitname| (exitname[,parameter])
                [,parameter,parameter,...])
WHERE - exitname = name of the user-written program to receive
                control for I/O-related requests
.      parameter = a valid parameter in the same format as those
                parameters which can be specified on the
                RUN TASK command.
```

The IOEXIT keyword is valid in either the FROM or TO areas of the COPY statement. This enables you to specify a different user-written I/O Exit on each side as illustrated in the following example.

COPY FROM	(PNODE,	-
	IOEXIT=(INEXT01,C'DB0A05',X'0E'))	-
TO	(SNODE,	-
	IOEXIT=OUEXT03)	

If an exit is specified, it can ignore the values of the other parameters on the COPY statement, such as the DCB information. This is beyond the control of the software. See the Connect:Direct Processes Web site at www.sterlingcommerce.com/Documentation/processes/processhome.html for information about the I/O Exit.

Specifying the I/O Exit in the TYPE File

Another method of indicating that an I/O Exit will be used is to include the IOEXIT keyword on the INSERT and UPDATE Type file commands. The format is the same as on the COPY statement. If an IOEXIT parameter is specified on the COPY statement, it overrides any IOEXIT specified in the type file entry. The type defaults record must reside on the side (source or destination) of the copy on which it was referenced.

I/O Exit Access to Control Blocks

On entry to the exit, register 1 (R1) contains the address of the pointer to the EXTTCB (Exit Control Block). As with other user exits, the parameter list addresses point to a two-byte length followed by the value.

All parameter lists end with the high order bit on in the last address in the list. The EXTTCB is generated by the macro DMEXITCB. The following is a listing of the I/O exit control block.

```

*****
*                               DMEXITCB VERSION 1 RELEASE 1 MODIFICATION 0           1991 *
*                               *
*   TYPE: Control Block           *
*   NAME: Exit Control Block      *
*                               *
*-----*
EXTCB      DSECT
EXTCBLNG  DC      Y(EXTCB#)      length of exit control block
EXTIDENT  DC      CL2' `        exit type identifier
*                               IO = I/O exit
*                               SQ = SQL exit
EXTFORW   DC      A(0)          forward chain pointer
EXTBACKW  DC      A(0)          backward chain pointer
EXTEXITN  DC      CL8' '        name of user exit program
EXTEPA    DC      A(0)          EPA of user exit program
EXTPARML  DC      A(0)          plist address for exit program
EXTOPER   DC      CL8' '        requested exit operation
EXTTASKN  DC      H'0'          task number
EXTRTNCD  DC      F'0'          return code from exit program
EXTRCOK   EQU     X'00000000'    - normal
EXTRCEOD  EQU     X'00000004'    - end of data
EXTRCDEL  EQU     X'00000008'    - delete record
EXTRCINS  EQU     X'0000000C'    - insert record
EXTRCLGC  EQU     X'FFFFFFFF'    - logic error
EXTMSGID  DC      CL8' '        message id from exit program
EXTWKARA  DC      A(0)          exit program work area address
EXTINARA  DC      A(0)          input record area
EXTOTARA  DC      A(0)          output record area
EXTINLNG  DC      F'0'          input record length
EXTOTLNG  DC      F'0'          output record length
EXTMAXLN  DC      F'0'          maximum output record length
EXTSRECF  DC      CL4' '        source recfm
EXTSBLKZ  DC      CL5' '        source blksize
EXTDRECL  DC      CL5' '        destination lrecl
EXTDRECF  DC      CL4' '        destination recfm
EXTDBLKZ  DC      CL5' '        destination blksize
EXTDIR    DC      C' '         direction of transfer
*                               S = sending side
*                               R = receiving side
EXTSPAN   DC      C' '         is record a spanned segment?
*                               Y = yes
*                               N = no
EXTFUNC   DC      C' '         requested function (DATAEXIT only)
*                               I = initialize
*                               P = process record
*                               E = exit
EXTSFLAG  DC      X'00'        source descriptor flags
EXTSRCL   EQU     X'80'        - source lrecl specified
EXTSRCF   EQU     X'40'        - source recfm specified
EXTSBKZ   EQU     X'20'        - source blksize specified
EXTSIOX   EQU     X'10'        - IOEXIT specified
EXTSSQL   EQU     X'08'        - SQL specified
EXTSDBP   EQU     X'04'        - DBPARMS specified

```

(continued)

EXTSDOCK	EQU	X'02'	- checkpoint negotiated
EXTSNOCK	EQU	X'01'	- checkpoint not performed
EXTDFLAG	DC	X'00'	destination descriptor flags
EXTDRCL	EQU	X'80'	- destination lrecl specified
EXTDRCF	EQU	X'40'	- destination recfm specified
EXTDBKZ	EQU	X'20'	- destination blksize specified
EXTDIOX	EQU	X'10'	- IOEXIT specified
EXTDSQL	EQU	X'08'	- SQL specified
EXTSRECL	DC	CL5'	source lrecl
EXTDDBP	EQU	X'04'	- DBPARMS specified
EXTDDOCK	EQU	X'02'	- checkpoint negotiated
EXTDNOCK	EQU	X'01'	- checkpoint not performed
EXTMISC	DC	X'00'	general flags for exit processing
EXTCONI	EQU	X'80'	- exit got ctl at least once
EXTCONP	EQU	X'40'	- exit got ctl but has not returned
EXTCONR	EQU	X'20'	- exit got ctl, returned at least once
EXTCKPT	EQU	X'10'	- exit should do checkpointing
EXTABN	EQU	X'08'	- exit should perform rollback
EXTBLK	EQU	X'04'	- exit sending/receiving blocks of data
EXTFLAG1	DC	X'00'	- more flags for exit processing
EXTCKPTA	EQU	X'80'	- exit is ready to do checkpointing
EXTIRECN	DC	F'0'	# records read from dataset (inrecn)
EXTIBLKN	DC	F'0'	# blocks read from dataset (inblk)
EXTORECN	DC	F'0'	# records written to dataset (otrecn)
EXTOBLKN	DC	F'0'	# blocks written to dataset (otblk)
EXTIBYTN	DC	D'0'	# bytes read from dataset (inbyte)
EXTOBYTN	DC	D'0'	# bytes written to dataset (otbyte)
EXTCKPT1	DC	XL16'00'	for checkpointing
EXTVSWRK	DC	A(0)	work area address (waadr)
EXTVSWKL	DC	F'0'	work area length (walen)

I/O Exit Requests

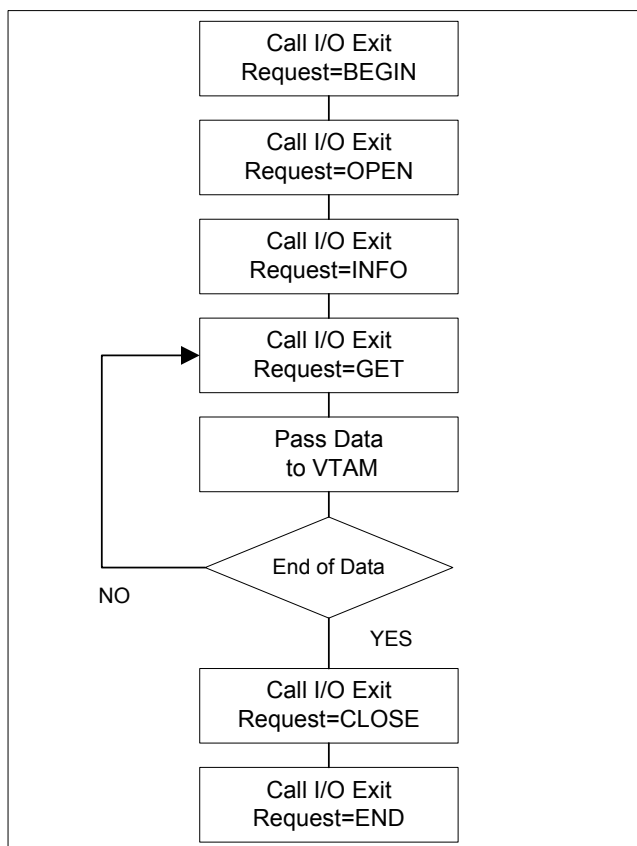
The I/O Exit will be called with Connect:Direct requests that are found in EXTOPER, which is a field in EXTCB. The following are the requests that the input and output I/O exits will receive.

Request	Description
BEGIN Request	The BEGIN request is made to an I/O Exit when the software wants to begin communication with the exit. The exit should take this opportunity to allocate work areas in preparation for future requests. This is the first request an I/O Exit will receive.
OPEN Request	The software sends an OPEN request to an I/O Exit when a file is to be allocated and opened. EXTDIR will contain either S or R to indicate whether the file is to be read or written. The I/O Exit should use EXTWKARA to anchor any storage obtained and set EXTMAXLN to the maximum record length.

Request	Description
INFO Request	<p>The INFO request is made to an I/O Exit when the software wants the exit to retrieve the file attributes and place them into the INFO area (mapped by the DMINFO macro) which is pointed to by EXTVSWRK. These data set attributes are required by the system. The following fields in the INFO control block should be set. The values listed are an example of those needed for a sequential data set.</p> <p>INBLKSZ = F'80' block size INLRECL = F'80' record size INTYPE = CL4'PS' data set organization INRECFM = CL4'RB CL4'VB' blank INUNIT = CL8' ' blank INBLKS = F'0' nulls INUSEBLK = F'0' nulls INBLKTRK = F'0' nulls INTRKCYL = F'0' nulls IN2NDRY = 8C'0' character zeros INLOCTYP = CL3' ' blanks</p>
COPY Routine Local Block/Deblocker	<p>By setting the EXTBLK switch on during the info coll, your I/O exit can use block mode transfers to/from destinations that require record mode transfers. If you are sending a file, your exit will send a complete block of data either FB or VB. If you are receiving a file, your exit will receive a complete block of data either FB or VB. COPY routine will either block or deblock the data.</p>
GET Request	<p>The GET request is made to an I/O Exit when the software wants a record/block to be read into the buffer. EXTINLNG should be set to the length of the data. EXTINARA should point at the record obtained. The Exit must indicate normal END-OF-DATA condition to the system by returning a value of EXTRCEOD in EXTRTNCD. Other conditions can be indicated by providing other values in the above mentioned fields. This allows messages that have been added to the message file to be issued. The IOEXIT must allocate a buffer for the input record/block. To determine if the Connect:Direct system is expecting a record or block, the IOEXIT must examine the source LRECL, BLKSIZE, RECFM and destination LRECL, BLKSIZE, RECFM fields in the EXIT control block. If the source and destination data set attributes match, then the software is expecting a physical block from the IOEXIT, otherwise a logical record is expected.</p>
ADD Request	<p>The ADD request is made to an I/O Exit when the Connect:Direct software wants a record/block to be inserted. EXTOTLNG is set to the length of the data. EXTOTARA points to the new record/block.</p>
CLOSE Request	<p>The CLOSE request is made to an I/O Exit when the Connect:Direct software requests the file to be closed. Since all I/O has been completed, errors returned by the exit on this request are ignored. The EXTABN flag is set on if the CLOSE request is due to abnormal termination.</p>
END Request	<p>The END request is made to an I/O Exit when the Connect:Direct software wants to end communication with the exit. The exit should take this opportunity to release any work areas it may have allocated when the BEGIN request was received. This is the last request an I/O exit receives.</p>

Normal Input Calling Sequence

The following figure illustrates the normal call sequence for an I/O Exit used for input.



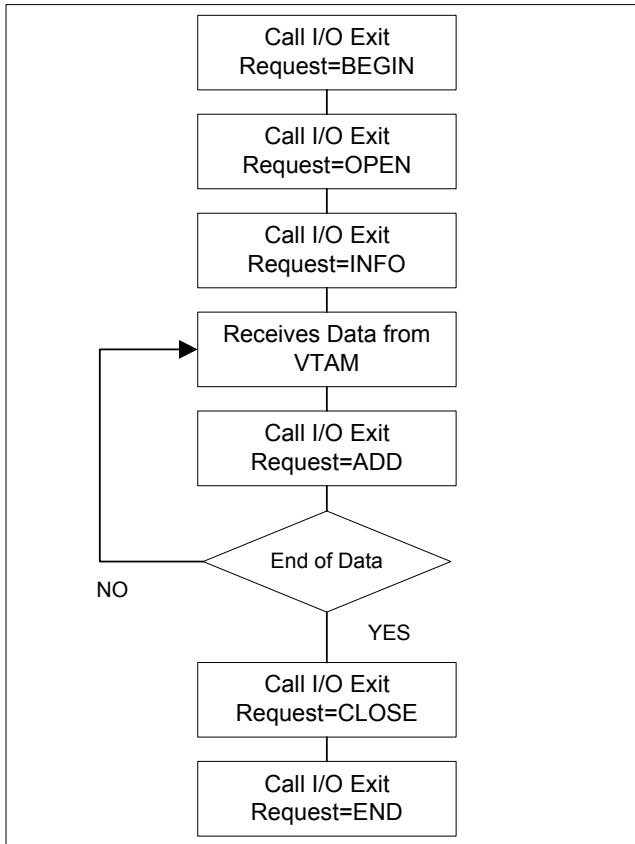
1. Connect:Direct calls the I/O Exit - request = BEGIN
2. Connect:Direct calls the I/O Exit - request = OPEN
3. Connect:Direct calls the I/O Exit - request = INFO
4. Connect:Direct calls the I/O Exit - request = GET
5. Connect:Direct passes data to VTAM

Calls 4 and 5 repeat until End of Data is returned by the I/O Exit.

6. Connect:Direct calls the I/O Exit - request = CLOSE
7. Connect:Direct calls the I/O Exit - request = END

Normal Output Calling Sequence

The following figure illustrates the normal calling sequence of an I/O Exit used for output.



1. The Connect:Direct software calls the I/O Exit - request = BEGIN
 2. The software calls the I/O Exit - request = OPEN
 3. The software calls the I/O Exit - request = INFO
 4. The software receives data from VTAM
 5. The software calls the I/O Exit - request = ADD
- Calls 4 and 5 repeat until there is no more data to be received from VTAM.
6. The software calls the I/O Exit - request = CLOSE
 7. DM calls the I/O Exit - request = END

Maintaining Connect:Direct for VSE/ESA

This chapter explains the steps required for updating the Connect:Direct for VSE/ESA software:

- ◆ Installing maintenance tapes
- ◆ Maintaining network map definitions
- ◆ Adding messages to the message library

Step 1 – Install Maintenance Tapes

Each maintenance tape is accompanied by *Connect:Direct for VSE/ESA Release Notes* which describes:

- ◆ Problems that were fixed
- ◆ Any enhancements
- ◆ The files on the tape
- ◆ Any special installation considerations

Execute the jobstream to unload the procedure library, as described in the *Step 2–Load MSHP Installation Libraries to Disk* section of the *Connect:Direct for VSE/ESA Installation Guide*. From this point the installation process is the same as a base installation except for the following:

- ◆ Do not delete or redefine the VSAM files when you run NDMINSTL. This can be accomplished by deleting the DELETE and DEFINE statements in the DEFVSAM jobstream for the VSAM files to be preserved or by eliminating the execution of the DEFVSAM procedure.
- ◆ Copy the phase DMGRATX from the original Connect:Direct for VSE/ESA sublibrary to the new Connect:Direct sublibrary after running the NDMINSTL jobstream. This is the Asset Protection module that works with your permanent Connect:Direct Key. This step can be disregarded for new installations.
- ◆ Always check the *Connect:Direct for VSE/ESA Release Notes* for exceptions to these rules.

Step 2 – Maintain Network Map Definitions

The network map file must be deleted, redefined and reloaded for any change to the network map. Refer to Chapter 5, *Defining and Maintaining the Network Map* for more information about defining the network map keywords LOCAL.NODE and ADJACENT.NODE.

Step 3 – Add Messages to Connect:Direct Message File

Users can load special messages into the message file. The following sections show how to load a message into the message file and a sample message source format.

Sample Format for Message Source

The sample format for the message source, shown in the following example, is in the MSGLOAD jobstream, found in the system procedure library. Copy this member into your message source library and change it as needed.

Rules for Variables and Message IDs

- ◆ The DELETE, INSERT, and MODULE variables can be 1-8 characters long.
- ◆ The STTEXT and L01 through L12 variables can be 1-64 characters long.
- ◆ To insert a message ID, the following are required: INSERT, MODULE, STTEXT, and L01 through L12.
- ◆ To delete a message ID, the following is required: DELETE.
- ◆ To replace a message ID, the following are required: DELETE, INSERT, MODULE, STTEXT, and L01 through L12.

Use the format in the following example exactly as shown. Comments are not allowed.

```
DELETE=MSG00001
INSERT=MSG00001
MODULE=MSGSOURC
STTEXT= This is an example of the short text message (one).
L01 = This is an example of the long text message (one). As many
L02 = as 12 lines can be used for the long text message.
L03 =
L04 =
L05 =
L06 =
L07 =
L08 =
L09 =
L10 =
L11 =
L12 =
```

Note: The STEXT and long messages are truncated to 63 bytes.

Jobstream to Update Message File

The jobstream used for updating messages, called MSGLOAD, is shown in the following example.

Using Statistics Administration

This chapter includes discussions of statistics administration. The topics covered are:

- ◆ Statistics facility overview
- ◆ Archiving statistics
- ◆ Monitoring the statistics facility

Statistics Facility Overview

The Connect:Direct for VSE/ESA statistics facility logs statistics to a series of pairs of VSAM files. Each pair consists of an entry sequenced file and a key sequenced file, both with the REUSE attribute. The minimum configuration is two file pairs, or four files. You can configure more file pairs than the minimum of two. You specify the file pairs in the initialization parameters STAT.DSN.BASE and STAT.FILE.PAIRS, as data set name high level qualifiers and the number of pairs. This determines the configuration of the statistics file pair list.

File Pairs

Within each file pair, Connect:Direct writes the statistics records to the entry sequenced file, while the key sequenced file maintains index information about the records. On average, Connect:Direct writes records to the key sequenced file at the rate of about one for every two records written to the entry sequenced file. When you issue SELECT STATISTICS commands, the system can locate the requested records quickly and efficiently by using the key sequenced file as an index to the entry sequenced file. All the file pairs defined to the DTF are available to SELECT STATISTICS command processing. Connect:Direct searches any file pair that may contain records that satisfy the SELECT STATISTICS command, not just the files currently being written.

Order

Connect:Direct writes the statistics records to the entry sequenced VSAM files in chronological order, starting at the beginning of the file and proceeding until the file or its paired key sequenced file is full. The oldest record is always at the beginning of the file and the newest record is last. The

system records each statistics record as a single VSAM record. The system does not compress the records or add control information. When a file pair is full, the system switches to the next in the sequence, and begins writing to it. The list of file pairs can be thought of as being processed in a circular fashion, or as being *wrapped*. When the last file pair in the list is full, the system wraps back to the first pair in the sequence.

Switching

In addition to switching to a new file when the active one becomes full, a site can specify in the initialization parameters that a switch occur at certain times of the day using the `STAT.SWITCH.TIME=` initialization parameter. For example, you can specify that statistics files switch at midnight every day so a site can limit each file pair to records from a single day. `Connect:Direct` also provides an API command that you can use to direct the DTF to perform a switch at any given time.

Once the system has written to all the pairs in the list, the system reuses the pairs. When a switch is made, the system closes the active pair and makes the pair with the oldest data the new active pair. When the system switches to a file, or a file becomes active, `Connect:Direct` for VSE/ESA does a VSAM RESET. This erases any records and index information in the active file. The system then writes new records starting at the beginning of the file.

Archiving Statistics

Archiving refers to the process of copying the records from a statistics entry sequenced cluster to another data set for long term storage. The output of this process is an archive statistics file. You can write the archive file to a VSAM entry sequenced cluster with the same characteristics as a statistics ESDS, or to a non-VSAM sequential file on DASD. You can also write it to a magnetic tape or a database table. The system does not store the statistics records in the statistics ESDS in any special format. The system records each statistics record as a VSAM record in an ordinary VSAM ESDS. `Connect:Direct` for VSE/ESA provides a means of issuing the `SELECT STATISTICS` command against archived statistics. To make archived statistics available to `SELECT STATISTICS`, you must put the archived statistics in the format of a statistics file pair. In other words, you must use the form of a VSAM entry sequenced cluster with a paired VSAM key sequenced cluster containing the index information. For example, if the records are archived to a magnetic tape file, you must first copy the archived records to a VSAM ESDS. Then you can run the `DMSTBKEY` utility to build the required associated VSAM KSDS. Refer to the discussion of estimating space requirements in the *Planning the Installation* chapter in the *Connect:Direct for VSE/ESA Installation Guide* for information about the characteristics and relative sizes of the keyed and entry sequenced clusters of a file pair. See page 132 for an explanation and example of the `DMSTBKEY` utility.

`Connect:Direct` provides a number of features for archiving statistics records. Using the DTF initialization parameter `STAT.SWITCH.SUBMIT`, you can specify that when the DTF switches from one statistics file pair to another, `Connect:Direct` submits a pre-defined Process to archive the records in the previously active ESDS. `Connect:Direct` submits this archive Process with a symbolic parameter indicating the data set name of the ESDS of the pair. The Process can then use

Connect:Direct to copy the data to another location. A sample archive Process is in the source library, member ARCHSTAT. Alternatively, the Process can submit a batch job to archive the data using IDCAMS REPRO, or some other utility. This can be done using the DMRTSUB utility to substitute the file data set name into the submitted jobstream. You can invoke DMRTSUB through the RUN TASK statement. A sample archive Process that submits a batch job using DMRTSUB is in the source library, member ARCHSTRJ. The job submitted is in the source library, member ARCHJOB.

The archive must complete before Connect:Direct needs to reuse the file being archived, that is, at the time of wrap-around of the file pair list. The completion of the archiving Process is important because Connect:Direct erases the contents of the statistics file when the system switches to that file. In other words, archiving must complete within the time required for the file pair list to wrap. Normally this should present no problems.

Several options are available to coordinate the archiving of statistical data with the reusing of the file pairs. The STAT.ARCH.CONFIRM initialization parameter specifies whether or not the site wants to ensure that data is archived before the system erases the file. If the site does not care about archiving, Connect:Direct simply resets the files when the switching occurs, and begins writing. If the site wants archiving, Connect:Direct verifies that the archive is complete before proceeding. In this case, Connect:Direct requires notification of archival. There are several ways to notify Connect:Direct.

If the archive is done using the COPY statement in a Connect:Direct Process, then the Process can also invoke the DMSTARRT utility when the COPY successfully completes. Connect:Direct invokes DMSTARRT through a RUN TASK statement, and notifies Connect:Direct that the data has been archived. If you use a batch job to archive, then the system can send the notification by including a step to execute the DMSTARBT utility. Also, you can issue the API command STATISTICS ARCHIVED to inform Connect:Direct to reuse a file pair. If there is no indication regarding the completion of the archive when Connect:Direct needs to reuse the files, the system issues a message to the operator console and waits for a reply indicating permission to reuse the file.

Note: In this situation, all activity in the DTF will cease until there is a response to the message from the operator indicating that the statistics file can now be written. This occurs as a result of the request that the DTF not erase statistics data unless it is certain that archiving the statistics is complete.

Connect:Direct does not require archive confirmation before reusing a statistics file pair when you specify or default to the DTF initialization parameter, STAT.ARCH.CONFIRM=NO. If you specify this initialization parameter, you are responsible for ensuring that the archive successfully completes before Connect:Direct resets the file. If the file is reset before copying the records, the data is lost. If the records are in the process of being copied when Connect:Direct needs to reset the file, then Connect:Direct must wait for the copy to complete. This is because Connect:Direct must have exclusive access to the file to do the VSAM reset. In this situation, Connect:Direct will also issue a message to the operator console and wait for a reply indicating that the file can be reset.

As mentioned previously, for archived statistics to be available to the SELECT STATISTICS command, the archived statistics must be in the form of a statistics file pair. Use the ARCHDSN parameter of the SELECT STATISTICS command to search archive files. The ARCHDSN parameter names only the key sequenced clusters of the archive pairs; Connect:Direct locates the associated entry sequenced clusters using control information in the key sequenced clusters.

Connect:Direct does not examine the statistics file pair list of the DTF when using the ARCHDSN parameter. Connect:Direct only searches the archive files. SELECT STATISTICS processing does not let you name files currently in the file pair list of the DTF in the ARCHDSN parameter or combine archive files with files in the file pair list. Refer to the *Examining Completed Process' Results and Statistics* chapter of the *Connect:Direct for VSE/ESA User's Guide* for a description of the SELECT STATISTICS command and details on how to specify the ARCHDSN parameter.

Statistics Archive File Directory

Connect:Direct also provides the capability of maintaining a directory of statistics archive files. The directory is a VSAM key sequenced file that contains a record for each archive file. Information in the record includes the data set name of the archive file and the range of dates and times covered by the archived records. Refer to the *VSAM Files Space Requirements* section of the *Planning the Installation* chapter in the *Connect:Direct for VSE/ESA Installation Guide* for an explanation of estimating space requirements allocating the directory file.

In order to use the directory feature, you must allocate the directory file and specify its name in the STAT.ARCH.DIR initialization parameter. Connect:Direct provides a means of viewing the directory contents in the form of the INQUIRE STATDIR command. Refer to *Displaying Statistics Archive File Directory* on page 21 for an explanation of the INQUIRE STATDIR command.

The archive notification utilities, DMSTARRT and DMSTARBT, write the directory records. If you wish to use the directory feature, you must execute one of these utilities from the Process or batch job that archives the records. This is true even if you do not specify STAT.ARCH.CONFIRM=YES in the DTF initialization parameters. You must also use these utilities to send archive notification when you are not using the directory feature, but specify STAT.ARCH.CONFIRM=YES in the initialization parameters file.

Archiving Related Utilities

The following section explains the archiving related utilities.

DMSTARRT

The DMSTARRT utility (S**T**atistics **A**Rchive **R**un **T**ask) has the following two functions:

- ◆ Notifies Connect:Direct of the availability of a statistics file pair for reuse due to the completion of archiving
- ◆ Optionally adds an entry to the directory of archive files

You can invoke DMSTARRT from within a Process by use of RUN TASK. You should use this utility when submitting a Process at statistics file pair switch time that archives the statistical data using Connect:Direct to COPY the statistics to another file. Once there is successful completion of the COPY, the system can update the directory and send the archive notification.

The program accepts two parameters through the RUN TASK statement. The first parameter is required and is the data set name of the statistics entry sequenced cluster that was archived. The second parameter is optional, and is the data set name of the archive file.

DMSTARRT *always* sends archive notification to the DTF. If you specify STAT.ARCH.CONFIRM=NO and there is no requirement of notification, then the notification simply has no effect.

The addition of the entry in the directory of archive files is done optionally depending on the specification of the second parameter string. If the second parameter is present, then the system updates the directory to contain an entry for the new archive file.

Sample Archiving Process Using DMSTARRT

The following is an example of an archive Process. This Process copies a statistics file to a sequential disk file and then invokes DMSTARRT to send archive notification to the DTF and update the directory of archive files. Connect:Direct passes the data set name of the statistics file to the Process in the form of the symbolic parameter &EDSN.

```

ARCPROC1 PROC  SNODE=SC.VSE.NODE                -
               &EDSN=,                          -
               &ODSN=CDVSE.TEST.STATARC1        -
               PRTY=15                            -
               STARTT=(TODAY)
*
STEP0001 COPY  FROM (                            -
               DSN=&EDSN                          -
               DCB=(DSORG=VSAM)                   -
               )                                   -
               TO (                               -
               DSN=&ODSN                          -
               UNIT=DLBLONLY                      -
               DCB=(DSORG=PS, RECFM=VB, LRECL=1604, BLKSIZE=28332) -
               )
*
STEP0002 IF    (STEP0001 EQ 0) THEN              -
               RUN TASK (PGM=DMSTARRT, PARM=( "&EDSN" ,
               "&ODSN" ) )
               EIF
STEP0003 IF    (STEP0001 EQ 0) THEN              -
               RUN TASK (PGM=DMNOTIFY,          -
               PARM=( 'GOOD' , &EDSN) )        -
               PNODE
               ELSE
               RUN TASK (PGM=DMNOTIFY,          -
               PARM=( 'FAIL' , &EDSN) )        -
               PNODE
               EIF

```

DMSTARBT

The DMSTARBT utility (S**TA**istics **A**Rchive **Ba**Tch) has the following two functions:

- ◆ Notifies Connect:Direct that a statistics file pair has been archived and is now available for reuse
- ◆ Optionally adds an entry to the directory of archive files

You should execute DMSTARBT as a step within a batch job. Use DMSTARBT when submitting a job that archives the statistical data by executing IDCAMS or some other utility to COPY the data

to another file at statistics file pair switch time. The system can update the directory and send archive notification once there is successful completion of the COPY.

DMSTARBT requires that the system allocate the archived statistics file with the data definition name (DLBL FILE NAME) of STESDS.

DMSTARBT *always* sends archive notification to the DTF. If you specify STAT.ARCH.CONFIRM=NO and there is no requirement of notification, then the notification simply has no effect.

If you want DMSTARBT to update the directory of archive files, then the system must allocate the following two files:

- ◆ STADIR - the directory file
- ◆ STARCH - the archive file

Sample Archiving Process Using DMSTARBT

In the following example, the archive Process submits a batch archive job using DMRTSUB to substitute the statistics file data set name into the jobstream. The system passes this data set name to the archive Process as the symbolic parameter &EDSN. Refer to the *Using the Utility Programs* chapter of the *Connect:Direct for VSE/ESA User's Guide* for details about using DMRTSUB.

The following figure is a sample archive Process.

ARCHSTRJ	PROCESS	PNODE=primary.node	-
		SNODE=secondary.node	-
		PRTY=10	-
		&EDSN=	-
RUN	TASK	(PGM=DMRTSUB, PARM= ("DSN=ARCHJOB"	-
		"DSNAME &EDSN")	

DMSTBKEY

The DMSTBKEY utility (Statistics Build KEYs) loads a statistics key sequenced cluster with index information for an associated statistics entry sequenced cluster. DMSTBKEY must execute as a batch job step.

DMSTBKEY allows the recreation of index information for archived statistics data so that you can issue a SELECT STATISTICS command. You can also use this utility to rebuild index information for statistics files in the DTF file pair list in certain cases. Refer to *Changing the File Pair Configuration* on page 139 for additional information.

DMSTBKEY requires the allocation of DLBL FILE NAMEs ESDSnn and KSDSnn with the entry sequenced and key sequenced clusters respectively. Connect:Direct loads the entry sequenced cluster with the statistics records for building the index information. The key sequenced cluster must either be empty, or be defined with the REUSE attribute. DMSTBKEY will erase any records in the KSDS before writing the new information. The size of the KSDS should be about 15% of the size (in records) of the associated ESDS. The KSDS must have the characteristics of a statistics key sequenced cluster. Refer to the *VSAM Files Space Requirements* section of the *Planning the Installation* chapter in the *Connect:Direct for VSE/ESA Installation Guide* for details about allocating statistics clusters.

Sample Jobstream to Execute DMSTBKEY

The following is an example of a jobstream to execute DMSTBKEY.

```
* $$ JOB JNM=DMSTRBKEY, CLASS=A, DISP=D, USER= ' STATBUILDKEY '
* $$ LST CLASS=Q, DISP=L, JSEP=1, LST=SYSLST
// JOB DMSTBKEY
// EXEC PROC=CDLIBS
// DLBL ESDS01, 'STAT.ESDS01', , VSAM
// DLBL KSDS01, 'STAT.KSDS01', , VSAM
// DLBL SYSOUT, 'SYSOUT.SYS004'
// ASSGN SYS004, SYSLST
// EXEC DMSTBKEY, SIZE=AUTO
/*
/&
* $$ EOJ
```

Sample Archiving Setup

Assume that a site has the following requirements for archiving:

- ◆ Statistics records must remain available for seven days in the file pair list before being overwritten by new records. After seven days, they must be available in archive files.
- ◆ Each archive file is to contain no more than one days statistics records.
- ◆ Batch jobs executing the IDCAMS utility to copy the records to sequential files on magnetic tape must perform the archiving. The archive files must be available for 365 days.
- ◆ Maintain a directory of archive files.
- ◆ Ensure that statistics data is not overwritten before being archived.
- ◆ Establish a procedure for making archived statistics available to the SELECT STATISTICS command.

Configuring the Statistics Facility

The following describes how the installation can configure the statistics facility to satisfy these requirements.

You must determine how to configure the statistics file pair list. The rate at which you log statistics records, availability of the statistics records for seven days before being overwritten, and each archive file containing no more than one days statistics records will determine the size and number of file pairs required.

The statistics records remaining available for seven days after being generated means that the total record capacity of all the entry sequenced statistics files should be seven times the average number of records generated daily. Run the SCCSTAT utility to determine how many records, on average, the system writes daily. See *Monitoring the Statistics Facility* on page 136 for more information about the SCCSTAT utility.

According to the sample requirements a single archive file will contain up to the same number of records as a single statistics entry sequenced cluster, and an archive file should contain no more than a days records; it follows that each ESDS should also hold about a days records. This in turn implies that there be seven statistics file pairs. Use the STAT.SWITCH.TIME initialization parameter to

initiate a file pair switch every day at midnight rather than depend on a file pair switch occurring regularly as a result of file pairs filling. To ensure that the switches do not happen before midnight as a result of a file pair becoming full, make each ESDS slightly larger than the daily requirement.

The site determines, using SCCSTAT, that the system writes statistics records at the rate of about 11,000 daily. Therefore, define seven file pairs each with a capacity of about 12,000 records (RECORDS(12000)). This implies that the associated key sequenced clusters be defined with RECORDS(9000). Refer to the *VSAM Files Space Requirements* section of the *Planning the Installation* chapter of the *Connect:Direct for VSE/ESA Installation Guide* for details about allocating statistics clusters.

RECORDS(365) will define the directory of archive files because the system will generate the archive files at the rate of one daily and will retain the files for one year.

SYSTEMS.NDM.STATS is the data set name prefix for the statistics clusters. The data set name of the archive directory is SYSTEMS.NDM.STATS.DIRECTRY. Member ARCHPROC in the Connect:Direct Process Library will contain the archive process that is submitted at file pair switch time.

The following initialization parameters would be necessary for the sample archive requirements:

```

STAT.DSNBASE = SYSTEMS.NDM.STATS    /* data set name base      */
STAT.FILE.PAIRS = 7                  /* number of file pairs    */
STAT.SWITCH.TIME = ( 00:00 )        /* switch at midnight      */
STAT.SWITCH.SUBMIT = ARCHPROC        /* archive proc            */
STAT.ARCH.DIR = SYSTEMS.NDM.STATS.DIRECTRY /* use directory          */
STAT.ARCH.CONFIRM = YES              /* be sure archive completes */

```

The archive process in the member ARCHPROC is as follows:

```

ARCHIVE PROCESS  &EDSN=,              - /* passed stats dsname    */
                  SNODE=NDM.PROD,    - /* PNODE=SNODE            */
                  PNODE=NDM.PROD,    - /*                          */
                  CLASS=1,           - /* lowest class           */
                  PRTY=15,           - /* highest priority       */
                  REQUEUE=YES        - /* requeue on error      */
/*
/* invoke DMRTSUB to submit the archive job...
/*
      RUN TASK (PGM=DMRTSUB,          - /* execute DMRTSUB,      */
               PARM=("DSN=ARCHJOB"  - /* jobstream to sub     */
                    "DSNAME &EDSN")) /* pass stat dsname     */

```

The archive jobstream in the Connect:Direct sublibrary ARCHJOB.J shown in the following figure:

```

* $$ JOB JNM=ARCHJOB,CLASS=A,DISP=L,USER='ARCHSTATS'
* $$ LST CLASS=Q,DISP=L,JSEP=1,LST=SYSLST
// JOB ARCHJOB
// EXEC PROC=CDLIBS
// DLBL INPUT,'&DSNAME',,VSAM          /* FROM DMRTSUB          */
// DLBL OUTPUT,'CD.ARCH.STATS',,VSAM    /* ARCHIVE STATS FILE   */
// EXEC IDCAMS,SIZE=AUTO
    REPRO INFILE(INPUT) OUTFILE(OUTPUT)
/*
* ***** NOTIFY CONNECT:DIRECT THAT THE FILE PAIR CAN BE REUSED, ***
* ***** AND UPDATE THE DIRECTORY OF ARCHIVE FILES                    ***
*
// EXEC PROC=CDLIBS
// DLBL STESDS,'&DSNAME',,VSAM          /* FROM DMRTSUB          */
// DLBL STDIR,'CD.STATS.DIR',,VSAM      /* STATS DIRECTORY      */
// DLBL STARCH,'CD.ARCH.STATS'         /* ARCHIVE OUTPUT       */
// DLBL SYSOUT,'SYSOUT.SYS004'
// ASSGN SYS004,SYSLST
// EXEC DMSTARBT,SIZE=AUTO
/*
/ &
* $$ EOJ

```

The decision at the site is to have the Connect:Direct administrator handle requests for access to archived statistics records. The submitted requests will specify a range of dates and times for the necessary records.

The administrator will issue the INQUIRE STATDIR command to determine which archive files contain records for the specified period. The administrator will run the following jobstream to create a usable archived statistics file pair for each located archive file. The first step creates the archive file and copies the record to it. The second step builds the index information.

```

ARCPROC2 PROC  SNODE=SC.VSE.NODE           -
                &EDSN=,                     -
                &ODSN=CDVSE.TEST.STATARC2    -
                PRTY=15                      -
                STARTT=(TODAY)
*
STEP0001 COPY  FROM (                       -
                DSN=&EDSN                    -
                DCB=(DSORG=VSAM)            -
                )                             -
                TO (                         -
                DSN=&ODSN                   -
                LABEL=(1,SL)               -
                UNIT=TNOASGN               -
                DCB=(DSORG=PS,RECFM=VB,LRECL=1604,BLKSIZE=28332) -
                )
*
STEP0002 IF    (STEP0001 EQ 0) THEN          -
                RUN TASK (PGM=DMSTARRT, PARM=("&EDSN", -
                "&ODSN"))
        EIF
STEP0003 IF    (STEP0001 EQ 0) THEN          -
                RUN TASK (PGM=DMNOTIFY,      -
                PARM=('GOOD',&EDSN))        -
                PNODE
        ELSE
                RUN TASK (PGM=DMNOTIFY,      -
                PARM=('FAIL',&EDSN))        -
                PNODE
        EIF

```

The archived statistics are now available and you can issue `SELECT STATISTICS` against the statistics by coding the name of the key sequenced file with the `ARCHDSN` parameter, as shown in the following figure.

```

SELECT STATISTICS WHERE
                (PNAME=USERPROC, ARCHDSN=(SYS.NDMARCH.Dyymmdd))

```

Monitoring the Statistics Facility

Connect:Direct provides the following means of monitoring various aspects of the statistics facility.

- ◆ `INQUIRE STATISTICS` command
- ◆ Type S2 statistics records
- ◆ `SCCSTAT` utility
- ◆ Type SS statistics records

INQUIRE STATISTICS Command

The INQUIRE STATISTICS command gives a *snapshot* of the status of the facility. See *Displaying Statistics Logging Facility Status* on page 20 for an explanation of the INQUIRE STATISTICS command. INQUIRE STATISTICS produces a report that includes the following:

- ◆ List of any currently EXCLUDEed record types
- ◆ File pair list configuration and indication of which file pair is active
- ◆ Date and time range covered by each file pair
- ◆ Size of each file
- ◆ Percent utilization of the entry sequenced files
- ◆ Count of SELECT STATISTICS commands active for each file pair
- ◆ Indication if logging is waiting for SELECT STATISTICS to finish so a file pair can be reset
- ◆ Indication if logging is waiting for archive to finish so a file pair can be reset
- ◆ Reason for the last switch from each file pair
- ◆ Most recent file access return code and message ID for each file
- ◆ Percent utilization of the nonactive key sequenced files
- ◆ Indication of whether archive notification has been received for the nonactive files

S2 Statistics Records

The S2 statistics records contain information about the statistics logging function. The system writes the records about once per hour when there is activity in the DTF. Each S2 record contains statistics about the period of time since the prior S2 record was written. The S2 statistics records include the following information:

- ◆ Beginning time and length of the period covered
- ◆ Count of records written in the period
- ◆ Count of ESDS control intervals written in the period
- ◆ Count of total bytes written to the ESDS
- ◆ Average statistics record length
- ◆ Average records per control interval
- ◆ Average ESDS writes per second
- ◆ Average KSDS writes per second
- ◆ Average logging service time
- ◆ Total waits for logging queue element
- ◆ Each indexed field including max keys and average keys per control interval

Use the TYPE parameter of the SELECT STATISTICS command to view the S2 records. The system writes the S2 records with the userid specified in the STAT.USER initialization parameter. If you code a unique ID for STAT.USER and you specify the USER parameter on the SELECT STATISTICS request, you will greatly reduce the search time. This is because the userid is an

indexed field. See the *Using Environment Commands* chapter of the *Connect:Direct for VSE/ESA User's Guide* for more information on how to use the SELECT STATISTICS command.

For example, if a site codes STAT.USER=stuser, then a SELECT STATISTICS request to display all S2 records would look like the following figure.

```
SELECT STATISTICS WHERE (USER=stuser, TYPE=(S2)) TABLE
```

SCCSTAT

Use the SCCSTAT utility to find out the rate at which the system generates statistics records. SCCSTAT also performs an analysis of the contents of the statistics file showing what percentage of the records are of each record type. This utility runs as a batch job step, and analyzes a single statistics entry sequenced file. The following is an example of a jobstream to execute SCCSTAT.

```
* $$ JOB JNM=SCCSTAT,CLASS=A,DISP=D,USER='DISPLAYSTATS'
* $$ LST CLASS=A,DISP=D,LST=SYSLST,JSEP=1
// JOB SCCSTAT DISPALAY STATISTICS INDEX INFORMATION
// EXEC PROC=NDMLIBS
// DLBL ESDS,'CD.STATS.ESDS01',,VSAM /* ENTRY SEQUENCED */
// DLBL SYSOUT,'SYSOUT.SYS004'
// ASSGN SYS004,SYSLST
// EXEC SCCSTAT,SIZE=AUTO
/*
/&
* $$ EOJ
```

SS Statistics Records

The SS statistics records contain information about SELECT STATISTICS processing. One SS record is written for each SELECT STATISTICS command that executes. The SS record includes information such as which index was used to search the files and how many requests were issued to the keyed and entry sequenced clusters. The record also includes how many records were examined and rejected.

Use the SELECT STATISTICS command with the TYPE parameter to view the SS record. See the *Using Environment Commands* chapter of the *Connect:Direct for VSE/ESA User's Guide* for more information on how to use the SELECT STATISTICS command. Using selection criteria with the SELECT STATISTICS request will improve the performance by efficiently locating the requested records. For example, you can include the ID of the user that issued the SELECT STATISTICS command or the approximate time the request was issued, using the STARTT, STOPT, and USER parameters. The following figure shows an example using this selection criteria.

```
SELECT STAT WHERE
      (TYPE=(SS) USER=USER1 STARTT=(,NOON) STOPT=(,13:00)) TABLE
```

Changing the File Pair Configuration

You must make changes to the statistics files or to the configuration of the file pair list while the DTF is not running. During DTF execution, the files remain allocated by Connect:Direct.

The restrictions that Connect:Direct places on changes to the configuration are for maintaining the integrity of the facility. At DTF initialization time, Connect:Direct checks the statistics files to ensure their usability, and the validity and accessibility of their data.

File Pair Change Restrictions

Connect:Direct generates the statistics file pair list from the initialization parameters STAT.DSN.BASE and STAT.FILE.PAIRS. Within each file pair, Connect:Direct verifies the appropriate sizing, relative to each other, of the entry sequenced cluster and the key sequenced cluster. If the file pair is not relatively sized, then Connect:Direct issues a warning message and initialization continues. If either of the files of a pair has data, Connect:Direct attempts to verify that the two files are actually a statistics file pair. Connect:Direct attempts to verify that the key sequenced file really does contain index information for the associated entry sequenced file. This verification is done using control information maintained in the key sequenced file. Connect:Direct keeps a control record in the KSDS which contains the data set name and the control interval size of the paired entry sequenced file. If this information does not match, statistics initialization fails.

There are a few implications of this verification. Changing the control interval size of the ESDS or renaming the clusters will cause initialization to fail because the control record in the KSDS no longer matches the files. There are two ways to resolve this statistics initialization failure. One is to use the DMSTBKEY utility to rebuild the key sequenced cluster. This will recreate the KSDS control record so that it now matches the new names or control interval size. In this way the records in the file pair remain accessible when the DTF is available again. The second solution is to empty the files. In this way the file pair is available for logging new records; however, the old records are no longer available. You may want to archive the files before emptying them.

Changing the size of a file pair should not be a problem. The sizes of both files of a pair should change together so that the relative sizes do not change. If the files are made larger, and the names remain the same, then copy the records from the old smaller entry and key sequenced clusters to the new larger ones. Use the same procedure to make a file pair smaller, if all the existing records from both files will fit into the smaller space of the new files. If the existing records do not fit, then the new smaller file pair must be left empty initially, and the old records will become unavailable.

File Pair List Changes

The Connect:Direct statistics facility processes the statistics file pair list in a circular, or *wrap around* fashion. The system maintains statistics records in strictly chronological order both within each file pair, and with regard to the file pairs in the list. At DTF initialization time, unless STAT.INIT=COLD is specified in the initialization parameters, Connect:Direct verifies that the order of the file pairs is valid. This is done by examining the date and time range of each non-empty file pair in the list. These must be in strictly ascending order through out the list with the exception of across the *wrap point*. Empty file pairs can appear anywhere in the list.

It may be useful from time to time to change the number of file pairs in the list. This is done by changing the STAT.FILE.PAIRS initialization parameter which specifies the number of file pairs. This has the effect of adding to or removing from the end of the list. Add empty file pairs to the

end of the list unless you specify `STAT.INIT=COLD`. You can remove records from the end of the list by reducing the `STAT.FILE.PAIRS` value. When you remove these records, they become unavailable and may in some cases leave *gaps* in the statistics data. You may want to archive these records before removing them.

Programming Interface to Connect:Direct for VSE/ESA

This chapter discusses how to interface with the Connect:Direct for VSE/ESA product. The following topics are covered:

- ◆ Interfacing with Connect:Direct
- ◆ Writing a user application program
- ◆ API high-level interface
- ◆ How DMCHLAPI interprets parameters
- ◆ DMCHLAPI return codes

Interfacing with Connect:Direct

The two main components are the data transmission facility (DTF) and the application program interface (API).

- ◆ The DTF performs actual data transmission
- ◆ The API is the set of rules that govern the interchange of requests and responses between the DTF and a Connect:Direct application program

Application Interface Program

The high-level application interface program, DMCHLAPI, serves as a communications vehicle between the API and any application program that supplies command strings for processing in a batch environment. You can write an application program following the rules described in the following section.

Connect:Direct Command Strings

You can also process Connect:Direct command strings in a batch environment through the Connect:Direct batch interface program DMBATCH. Both DMCHLAPI and DMBATCH are

supplied during the installation. For more information on using DMBATCH, refer to the *Using Utility Programs* chapter of the *Connect:Direct for VSE/ESA User's Guide*.

Writing a User Application Program

User-written applications can be designed to interface with Connect:Direct. The applications can be written in any computer language, including PL/I, Assembler, and COBOL.

A sample of an Assembler language user written application, called ASMSAMP, can be found in the source distribution library.

A sample of a PL/I language user written program, called PLISAMP, can be found in the source distribution library.

API High-Level Interface (DMCHLAPI)

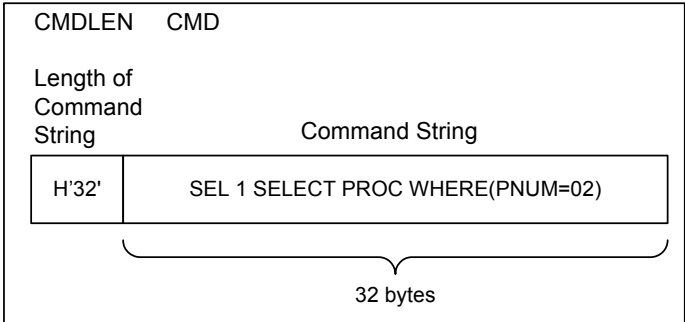
The high-level Connect:Direct interface program, DMCHLAPI, communicates with the API through a control block interface called the User Interface Control Block (UICB).

DMCHLAPI works in the following sequence:

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

Completing the Required Parameters

DMCHLAPI requires three parameters to be passed on every invocation from an application program:

Parameter	Description																											
CMDLEN	<p>(first parameter) is a halfword field that contains the length of the Connect:Direct command string. The command string, which can be from 1 to 1022 bytes long, is labeled CMD. The layout of the length and the command string is shown in the following figure.</p> 																											
UICB@	(second parameter) is a fullword pointer to UICB that is initialized at signon, and cleared by the SIGNOFF command.																											
OUTSPECS	<p>(third parameter) is a seven-byte character string, containing the output format specifications. Each specification is one character. See the following table for the valid values and meaning of each specification. (The output is written to the filename defined by DMPRINT.)</p> <table border="1"> <thead> <tr> <th>Field</th> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td rowspan="2">1</td> <td>Y</td> <td>Displays the command string that was executed</td> </tr> <tr> <td>N</td> <td>Does not display the command string that was executed</td> </tr> <tr> <td rowspan="2">2</td> <td>Y</td> <td>Displays the API return code and message ID</td> </tr> <tr> <td>N</td> <td>Does not display the API return code and message ID</td> </tr> <tr> <td rowspan="3">3</td> <td>S</td> <td>Displays the short message text when there is a return code of zero from the API</td> </tr> <tr> <td>L</td> <td>Displays long message text when there is a return code of zero from the API</td> </tr> <tr> <td>N</td> <td>Does not display message text when there is a return code of zero from the API</td> </tr> <tr> <td rowspan="3">4</td> <td>S</td> <td>Displays the short message text when there is a non-zero return from the API</td> </tr> <tr> <td>L</td> <td>Displays the long message text when there is a non-zero return code from the API</td> </tr> <tr> <td>N</td> <td>Does not display message text when there is a return code greater than zero from the API</td> </tr> </tbody> </table>	Field	Value	Meaning	1	Y	Displays the command string that was executed	N	Does not display the command string that was executed	2	Y	Displays the API return code and message ID	N	Does not display the API return code and message ID	3	S	Displays the short message text when there is a return code of zero from the API	L	Displays long message text when there is a return code of zero from the API	N	Does not display message text when there is a return code of zero from the API	4	S	Displays the short message text when there is a non-zero return from the API	L	Displays the long message text when there is a non-zero return code from the API	N	Does not display message text when there is a return code greater than zero from the API
Field	Value	Meaning																										
1	Y	Displays the command string that was executed																										
	N	Does not display the command string that was executed																										
2	Y	Displays the API return code and message ID																										
	N	Does not display the API return code and message ID																										
3	S	Displays the short message text when there is a return code of zero from the API																										
	L	Displays long message text when there is a return code of zero from the API																										
	N	Does not display message text when there is a return code of zero from the API																										
4	S	Displays the short message text when there is a non-zero return from the API																										
	L	Displays the long message text when there is a non-zero return code from the API																										
	N	Does not display message text when there is a return code greater than zero from the API																										

Parameter	Description	
5	Y	Displays the data that was generated in the temporary file
	N	Does not display the data that was generated in the temporary file
6	Y	Displays the string that identifies UICB fields about which the Extract feature will return information
	N	Does not display the string that identifies UICB fields
7	Y	Displays a dump of the area that received output from the Extract feature
	N	Does not display a dump of the area that received output from the Extract feature

All output generated as a result of these specifications is routed to a DLBL NAME of DMPRINT. No output is created if the DLBL NAME is not present.

The following example shows the most common specifications for this parameter.

```
OUTSPECS  DC      C 'YYSLYNN'
```

Positions 6 and 7 are not used by DMCHLAPI unless the extract feature of DMCHLAPI is used, as explained in the next section.

Using UICB Extract Feature

If the return code from the processed command is desired within the user program, the extract feature must be used. This allows extraction of certain UICB fields after command execution.

To activate the optional extract feature, the program must pass the following additional parameters to DMCHLAPI: EXTSTRLN, EXTAREA, EXTRC, EXTMSG, and INVALKEY.

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

```

/*****
/*      THE PARAMETERS WOULD BE DEFINED AS FOLLOWS:      */
/*****
CMDLEN  DS  H
CMD      DS  CL1022
UICB@   DS  A
OUTSPECS DC  C'YYSLYYY'
EXTSTR  DC  C`UITMPDDN UIESF UIPROC# UIMSGID'
EXTAREA DS  0CL24          /*OUTPUT FROM THE UICB EXTRACT      */
EXTMPDDN DS CL8          /*VALUE OF UITMPDDN          */
EXESF   DS  F            /*VALUE OF UIRTNCD          */
EXPROC# DS  XL4          /*VALUE OF UIPROC#          */
EXMSGID DS  CL8          /*VALUE OF UIMSGID          */
EXTRC   DS  F
EXTMSG  DS  CL8
INVALKEY DS CL8
/*****
/* NOTE THAT THE EXTRACT STRING ITSELF IS 30 BYTES LONG, BUT THE *
/* DUMP OF ITS EXTRACT WILL USE 31 BYTES                          *
/*                                                                *
/* INSIDE THE MAIN BODY OF THE PROGRAM, THE CALL TO DMCHLAPI WOULD *
/* LOOK LIKE:                                                    *
/*                                                                *
/*****
CALL DMCHLAPI, (CMDLEN,UICB@,OUTSPECS,EXTSTRLN,EXTAREA,
EXTRC,EXTMSG,INVALKEY),VL

```

The following is a list of optional parameters:

Parameter	Description
EXTSTRLN	<p>(fourth parameter) is the UICB extract string. The user-supplied application must set up an extract string identifying the UICB fields to be extracted. The string consists of a halfword length field containing the length of the extract string (excluding the halfword length field), followed by:</p> <ul style="list-style-type: none"> ◆ the first UICB field to be extracted ◆ a space ◆ the second UICB field to be extracted <p>The following is a sample extract string.</p>

H'32'	UITMPDDN UIRTNCD UIPROC# UIMSGID
-------	----------------------------------

Note: Only the UICB fields listed in the *UICB Field Listing* section later in this chapter can be specified in the extract string. If an invalid keyword is encountered in the string, the extract routine terminates execution, and no more information is extracted. The application is informed of the error by means of the EXTRC, EXTMSG, and INVALKEY parameters. If field six of the OUTSPECS parameter has been set to Y, the extract string is written to the file defined by DMPRINT.

Parameter	Description															
EXTAREA	<p>(fifth parameter) is where UICB extracted information is placed. The application is responsible for ensuring that the area is large enough to accommodate the information requested by the extract string. If field seven of OUTSPECS has been set to Y, the extract string is written to the file defined by DMPRINT.</p> <p>The following table shows how to calculate required storage for the extract area necessary for the extract string in the previous figure.</p> <table border="1"> <thead> <tr> <th>UICB Field Storage</th> <th>Data Definition</th> <th>Required Storage</th> </tr> </thead> <tbody> <tr> <td>UITMPDDN</td> <td>Character length</td> <td>8 bytes</td> </tr> <tr> <td>UIRTNCD</td> <td>Fullword</td> <td>4 bytes</td> </tr> <tr> <td>UIPROC#</td> <td>Fullword</td> <td>4 bytes</td> </tr> <tr> <td>UIMSGID</td> <td>Character length</td> <td>8 bytes</td> </tr> </tbody> </table>	UICB Field Storage	Data Definition	Required Storage	UITMPDDN	Character length	8 bytes	UIRTNCD	Fullword	4 bytes	UIPROC#	Fullword	4 bytes	UIMSGID	Character length	8 bytes
UICB Field Storage	Data Definition	Required Storage														
UITMPDDN	Character length	8 bytes														
UIRTNCD	Fullword	4 bytes														
UIPROC#	Fullword	4 bytes														
UIMSGID	Character length	8 bytes														
EXTRC	(sixth parameter) is a four-byte binary field containing the extract feature return code.															
EXTMSG	(seventh parameter) is an 8-character field containing an 8-character message ID from the extract feature.															
INVALKEY	(eighth parameter) is an 8-character field containing an invalid extract string keyword. DMCHLAPI relays the location of the extract string error to the application in this parameter. This field contains the invalid item in the extract string if EXTRC is non-zero.															

UICB Field Listing

The following four tables show the valid UICB fields. Specifying these fields in the EXTSTRLN parameter results in feedback in the extract area, defined by the EXTAREA parameter.

Exact definitions can be found in the DMUICB macro found in source library. The field names which appear boldfaced are explained further in the *DMCHLAPI Return Codes* section.

Name	Data Type	Length	Description
UIRCBLNG	halfword	2	Control block length
UIDESCR	character	16	UICB identifier
UITCA	address constant	4	Task Control Area
UIBRCB	address constant	4	Batch Region Control Block
UIDYNCB	address constant	4	Dynamic Allocation Control Block
UITPCB	address constant	4	Text Parser Control Block
UILEVEL	fullword	4	Modal level counter
UITMPDCB	address constant	4	Temporary file DCB

Name	Data Type	Length	Description
UIMSGCB	address constant	4	Message Control Block
UIUNODE	character	16	User node ID
UIUID	character	8	User ID
UIPSWD	character	8	Signon password
UINPSWD	character	8	New Signon password
UITMPVOL	character	6	Volume serial number of temporary file if not specified by user
UITMPDDN	character	8	Always NDMX001
UIUSRTYP	character	1	User Type (operator, administrator,user)
UIAPPLID	character	8	VTAM logon ID
UIRTNCD	fullword	4	API return code
UIMSGID	character	8	API message ID
UILNODE	character	16	Name of node that is "local" to DTF
UILPP	halfword	2	Lines per page for printed output
UITMPLNG	halfword	2	Length of temporary file name
UITMPDSN	character	44	Temporary file-ID (character length 44)
UIPUBLNG	halfword	2	Length of Process Library name
UIPUBDSN	character	44	Process Library name
UIMSGLNG	halfword	2	Length of message library name
UIMSGDSN	character	44	Message library name
UINETMAP	character	64	Network Map file name
UIPROC#	fullword	4	Process number from submit
UIDSPY	address constant	4	Address of SCDSPLY
UISTRING	address constant	4	Address of SCSTRING

Control Block Builder Syntax Error Work Areas

Each of the following will have a value only when appropriate.

Name	Data Type	Length	Description
UILABL#	halfword	2	Length of label
UILABL	character	8	Label

Name	Data Type	Length	Description
UICMD1#	halfword	2	Length of first word in command
UICMD1	character	8	First word in command
UICMD2#	halfword	2	Length of second word in command
UICMD2	character	8	Second word in command
UIKLST#	halfword	2	Length of keyword that starts a list
UIKLST	character	8	Keyword that starts a list
UIKEYW#	fullword	4	Length of keyword in list before error
UIKEYW	character	8	Last keyword in list before an error
UIPARM#	halfword	2	Length of parameter associated with UIKEYW
UIPARM	character	8	Parm in error associated with UIKEYW
UIERRM1#	halfword	2	Length of UIERRM1 string
UIERRM1	character	64	All of the above work areas resolved into a string
UIERRM2	character	64	Msg ID and text for parsing error in UICMD2

Start of Boolean Flags

Output is **Y** or **N** indicating bit is *on* or *off*.

Name	Data Type	Length	Description
UIERRON	character	1	Indicates message in UIERRM1
UIERRLAB	character	1	Indicates something in UILABL
UIERRCM1	character	1	Indicates something in UICMD1
UIERRCM2	character	1	Indicates something
UIERRSCP	character	1	Indicates parsing error
UIERRC10	character	1	open delimiter after command keyword, for example, "IF ("
UIERRC20	character	1	Indicates open delimiter after second command and keyword
UIERRG	character	1	Indicates VTAM error msg in UIERRM1
UIERRLST	character	1	Indicates something in UIKLST
UIERRLOP	character	1	Indicates open delimiter after UIKLST
UIERRLCL	character	1	Indicates close delimiter after UIKLST
UIERRLEQ	character	1	Indicates equal sign after UIKLST

Name	Data Type	Length	Description
UIERRLCM	character	1	Indicates comma after UIKLST
UIERRLSP	character	1	Indicates space after UIKLST
UIERRKEY	character	1	Indicates something in UIKEYW
UIERRKOP	character	1	Indicates open delimiter after UIKEYW
UIERRKCL	character	1	Indicates close delimiter after UIKEYW
UIERRKEQ	character	1	Indicates equal sign after UIKEYW
UIERRKCM	character	1	Indicates comma after UIKEYW
UIERRKSP	character	1	Indicates space after UIKEYW
UIERRBPC	character	1	Indicates close delimiter before a parameter in a list
UIERRPRM	character	1	Indicates something in UIPARM
UIERRPOP	character	1	Indicates open delimiter after UIPARM
UIERRPCL	character	1	Indicates close delimiter after UIPARM
UIERRPEQ	character	1	Indicates equal sign after UIPARM
UIERRPCM	character	1	Indicates comma after UIPARM
UIERRPSP	character	1	Indicates space after UIPARM
UIFSUBM	character	1	Indicates a Process was submitted
UIGOTDSN	character	1	Indicates found temporary file name as Signon command parameter
UITFILE	character	1	Indicates data was generated into temporary file
UIEOF	character	1	Indicates reached EOF of Process file
UIMODAL	character	1	Indicates modal statement processed
UITFILEX	character	1	Reserved
UIMASTER	character	1	Indicates this UICB is master
UIRECON	character	1	Indicates a reconnect attempted
UIINACT	character	1	Indicates VTAM session for this UICB failed
UILOCAL	character	1	Indicates local node session
UIESF	character	1	Indicates ESF=YES on Signon command
UIZOPSWD	character	1	Indicates blank password on Signon
UIZNPSWD	character	1	Indicates blank new password on Signon

The following table lists additional UICB Field Listings.

Name	Data Type	Length	Description
UIRAT@	address constant	4	PTR to resource address table
UIDRLSE	halfword	2	DTF release level
UIDPUF#	halfword	2	DTF PUF level
UILINE#	halfword	2	Line number within Process in error
UIRLSE#	halfword	2	Current release, version and mod level
UIPUF#	halfword	2	Current PUF level
UI@MASTR	address constant	4	Master (user's) UICB
UI@ACTIV	address constant	4	Currently active UICB
UI@FPTR	address constant	4	Next UICB
UI@BPTR	address constant	4	Previous UICB
UIALOTYP	character	8	Allocation type for temporary file
UIALOPRI	character	8	Allocation of prime space for temporary file
UIALOSEC	character	8	Allocation of secondary space for temporary file
UIALOUNI	character	8	Allocation unit for temporary file
UIALOVOL	character	8	Allocation of volume serial number for temporary file
UIFOLD	character	3	Fold to upper case if 'YES' UIPRTALC
UIPRTALC	character	80	User-defined print destination
UIPACCT#	halfword	2	Length of PNODE accounting data
UIPACCT	character	255	PNODE accounting data text
UISACCT#	halfword	2	Length of SNODE accounting data
UISACCT	character	255	SNODE accounting data text

How DMCHLAPI Interprets Parameters

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

DMCHLAPI Return Codes

The return codes shown in the following table reflect the status of the DMCHLAPI communications with the API. These codes only reflect whether DMCHLAPI could process the request and pass the command to the DTF. They do not reflect the completion status of the command.

Return Code	Meaning
00000000	The command executed normally
00000004	Signon to the master session failed, but the Connect:Direct Extended Submit Feature (ESF) environment was established
00000008	A non-ESF command attempted in an ESF environment
0000000C	A session was lost in a multiple session environment
00000010	The master session was lost
00000014	The master session was signed off successfully
00000018	The master session signon failed, and there is no ESF ability
0000001C	A non-master signon failed
00000020	DMCHLAPI received an invalid number of input parameters
00000024	The output specifications included an invalid parameter
00000028	An invalid pointer to the UICB was passed to DMCHLAPL

Extract Function Fields of Interest

The return feedback from the DTF regarding the completion status of the command must be obtained through the Extract function. Return codes received through the function inform you of an error or information generated by the requested command. Three particular fields of interest in the Extract function, applicable to DMCHLAPI return codes, are:

Field	Description
UIRTNCD	lists return codes. UIRTNCD is set on completion of every command processed by the DTF.
UIMSGID	lists message IDs. UIMSGID is set on completion of every command processed by the DTF.
UIPROC#	identifies process numbers. This is set for every successful SUBMIT command.

Examples of DMCHLAPI Parameters and Calling Sequences

The following examples show how the required and optional parameters and calling sequences might appear in assembler and PL/I applications.

DMCHLAPI Parameters for an Assembler Program

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

```

*****
*   THE DATA DEFINITIONS ARE AS FOLLOWS:   *
*****
CMDLEN  DS  H           LENGTH OF COMMAND STRING
CMD     DS  CL1022      COMMAND STRING
UICB@   DS  A           POINTER TO UICB
OUTSPECS DC  C'YYSLYNN' OUTPUT SPECIFICATIONS
*****
*
*   THE DMCHLAPI INVOCATION WOULD BE:       *
*
*****
CALL DMCHLAPI, (CMDLEN,UICB@,OUTSPECS),VL

```

DMCHLAPI Parameters for a PL/I Program

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

```

/*****
*   THE DATA DECLARATIONS ARE AS FOLLOWS:   *
*****
CMD_PARM FIXED BINARY (31) BASED (R)
UICB_PARM  FIXED BINARY (31) INITI (0'B')
OUTSPEC_PARM CHAR (7) INIT ('YYSLYNN')
CMD        CHAR (1022) VARYING,
R          POINTER;
/*****
*
*   THE DMCHLAPI INVOCATION WOULD BE:       *
*
*****
R=ADDR(CMD) ;
CALL DMCHLAPI(CMD_PARM, UICB-PARM, OUTSPEC_PARM);

```

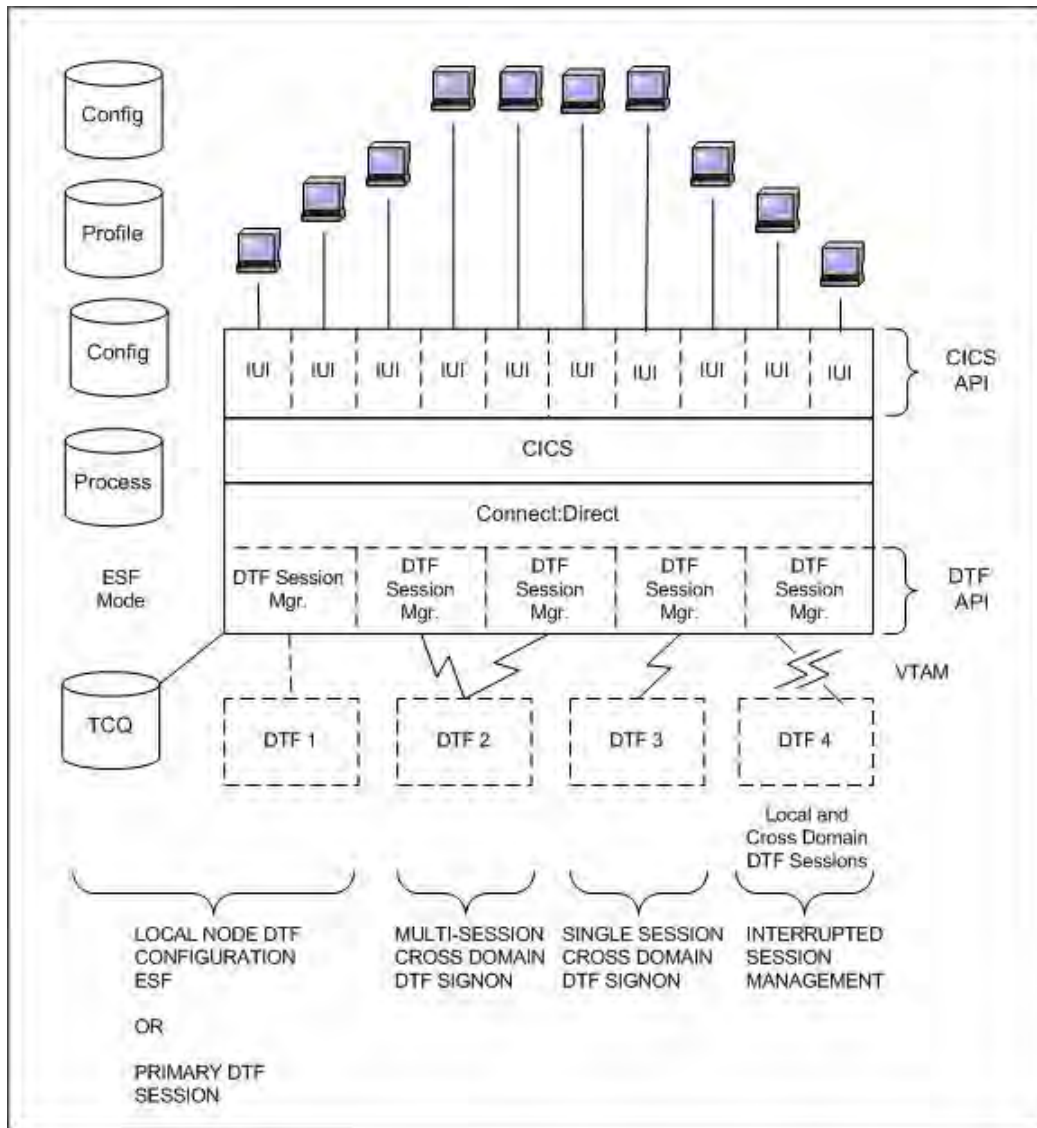
About the Connect:Direct CICS Interface Administration System

The Connect:Direct CICS Interface provides a number of components that enable users and applications to access, control, and transfer data across networks. The major components include:

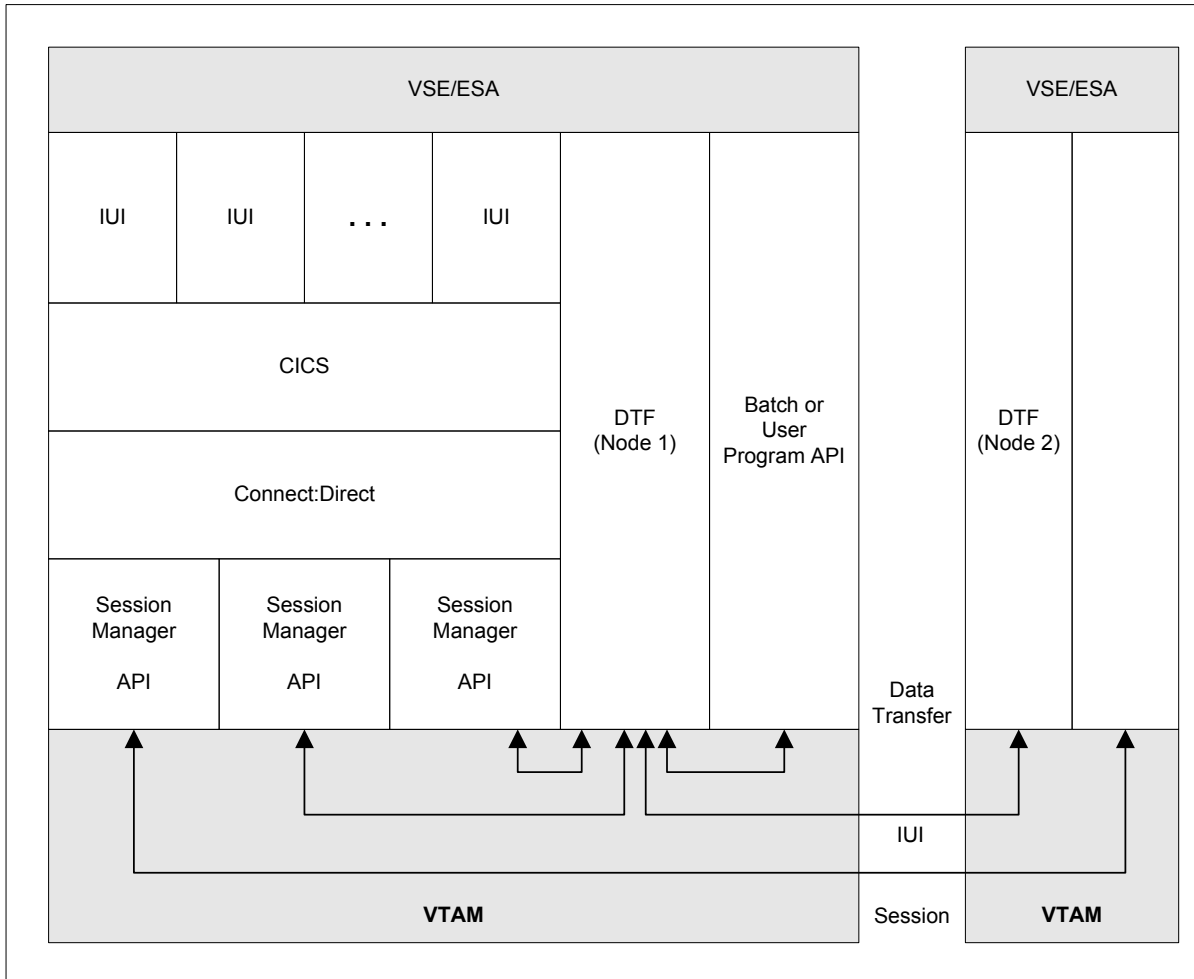
- ◆ **Interactive User Interface (IUI)** – designed with menus and data entry screens to enable users to transfer files, initiate applications, and monitor activity in a user-friendly environment.
- ◆ **Application Program Interface (API)** – allows the IUI and CICS Administration to communicate to the Data Transmission Facility (DTF) through the session manager. The API interprets the commands, but it is the responsibility of the session manager to establish communication sessions and perform standard session management functions.
- ◆ **Data Transmission Facility (DTF)** – the component that actually controls information distribution to other nodes in the network.
- ◆ **Extended Submit Feature (ESF) Mode** – allows data to be submitted for transmission even if the DTF is not active. Connect:Direct requests to an inactive node are routed to the sending Transmission Control Queue (TCQ) until the inactive node is initiated.

The following figure illustrates possible DTF sessions that can be established using Connect:Direct CICS Interface. These include the following:

- ◆ Connect:Direct DTF 1 - Local node DTF configuration ESF or primary DTF session.
- ◆ Connect:Direct DTF 2 - Multisession cross domain DTF signon.
- ◆ Connect:Direct DTF 3 - Single session cross domain DTF signon.
- ◆ Connect:Direct DTF 4 - Interrupted session management.



The following figure shows a general illustration of data transfer using Connect:Direct CICS Interface between two VSE/ESA operating systems.



Introduction to the Administration System

The Connect:Direct CICS Interface Administration System has been designed to help you with the configuration and control of the interface. The tasks you can perform while using this system include:

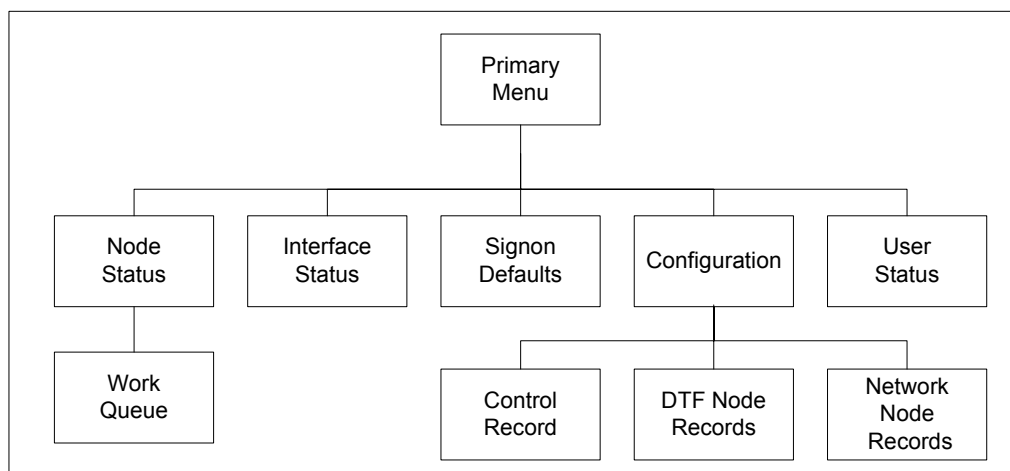
- ◆ Configuring Connect:Direct CICS Interface to provide the desired functionality to your users while maintaining optimum performance
- ◆ Defining the characteristics of the Data Transmission Facilities (DTFs) to which Connect:Direct can submit requests
- ◆ Activating and deactivating the interface

- ◆ Activating and deactivating the interface with specific DTFs
- ◆ Adding, modifying, and deleting default user signon characteristics
- ◆ Monitoring user activity and intervening if those activities compromise Connect:Direct or the CICS environment

The system guides you through a series of nested menus, prompts you for any needed information and then performs the requested function. If errors occur during the processing of your request, the system will inform you of the cause of the error and, in some cases, suggest remedial action.

Structure of the Administration System

The menu structure of the Connect:Direct CICS Interface Administration System is shown in the following figure.



By traversing these menus and screens and by entering the correct data, you can perform the following tasks:

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

Administration Screen Features

All the screens are based on the 24 X 80 (V X H) characters of a standard IBM 3270 Model 2 display.

System Fields

Each screen in the administration system contains information fields of various widths and attributes for your entered data and for system-displayed data. For those information fields that are for entered data, all variable data is highlighted. Required fields are highlighted and optional fields are not. Variable fields are underscored to indicate the size of the field.

As you perform your administrative activities, several system fields – DATE, TIME, and MESSAGE – are displayed with current information about your entered data, and are updated with system information and responses to your requests.

The 10-character DATE field contains the system date expressed as month, day, and year (MM/DD/YYYY), and occurs in the upper right corner of the screen.

The eight-character TIME field contains the system time expressed as hours, minutes, and seconds (HH:MM:SS), and occurs in the upper right corner of the screen.

The 75-character MESSAGE field occurs near the bottom of the screen, and contains the system messages associated with the success or failure of your administrative activities.

Connect:Direct CICS Interface messages have been added to the existing message library. The messages use the prefix SCCS to distinguish them from non-CICS messages. The format is *SCCSNNX*, where *NNN* is the number of the message, and *X* indicates I for information. Messages that can occur during your administration operations are listed in the *Connect:Direct CICS Interface Administration Messages* appendix in this manual.

PF Keys

Some menus have available programmable function (PF) keys for additional options and maneuvering through the menu hierarchy. These PF key assignments are displayed across the bottom two rows of the administration screens and function as follows:

Key	Function
PF3	Exits the current screen, and takes you to the previous screen
PF5	Adds an entry to the list on the screen
PF6	Deletes an entry from the list on the screen
PF7	Scrolls backward through the list of available data on the screen
PF8	Scrolls forward through the list of available data on the screen
PF9	Applies updates to screen information

Key	Function
PF10	Scrolls to the left on the screen
PF11	Scrolls to the right on the screen
ENTER	Refreshes screen data or processes a line command
CLEAR	Resets the data on the screen to default values

Using the Connect:Direct CICS Interface Administration System

This chapter explains the administration system and details the basic operation, including the use of the screens required. The chapter also details several operational considerations.

Signing On and Off

In order to use the administration system, you must have both CICS and Connect:Direct installed and working on your mainframe, and you must be using an IBM 3270 terminal or equivalent. Then you can sign on to the administration system.

Use these two steps to sign on to the Connect:Direct CICS Interface:

1. SIGNON to CICS.
2. SIGNON to Connect:Direct CICS Interface, using the CDA transaction.

Use the following steps to sign off the Connect:Direct CICS Interface:

1. Press **PF3** repeatedly until you reach the PRIMARY MENU, or press **PF4** once from your current screen to go to the PRIMARY MENU.
2. Press **PF3** once more, and you go to a blank screen with the message – CDA TRANSACTION ENDED – in the upper left corner.
3. Type in CSSF LOGOFF, and press **ENTER**.
4. View the message, DFH2206I - SIGN-OFF IS COMPLETE. This message displays briefly.

Using the Primary Menu

The Connect:Direct CICS Interface Administration PRIMARY MENU is the root of the administration menu hierarchy and is the access key to all the other features of the Administration System.

The PRIMARY MENU is shown in the following figure.

```

Connect:Direct for CICS ADMINISTRATION                                10:17:29
                          PRIMARY MENU

OPTION ==> _

C ... CONFIGURATION
I ... INTERFACE STATUS
N ... NODE STATUS
S ... SIGNON DEFAULTS
U ... USER STATUS
    To restrict U :
    CICS USERID ==> _____
    CICS TERMID ==> _____
Connect:Direct NODE ==> _____

PF keys:  3 Exit

```

Entry Fields

Field	Description
OPTION	contains the C, I, N, S, or U options. This field is one character long.
C	Selecting this option takes you to the CONFIGURATION menu. If you need to maintain configuration information, then select this option. This option is documented in Chapter 13, <i>Configuring the Connect:Direct CICS Interface Administration System</i> .
I	Selecting this option takes you to the INTERFACE STATUS menu. If you need to activate or deactivate the Connect:Direct interface, then select this option.
N	Selecting this option takes you to the NODE STATUS menu. If you need to view the status of nodes that users of Connect:Direct can sign on to, and information about the sessions and tasks under the node, then select this option.
S	Selecting this option takes you to the SIGNON DEFAULTS menu. If you need to maintain user signon default information, then select this option.
U	Selecting this option takes you to the USER STATUS menu. If the administrative user needs to check on a particular USERID, and the Connect:Direct activities associated with that USERID, or if the administrative user needs to cancel a user task or signon, then select this option.
Note: The following fields are not required under option U , but can be used to limit the scope of the display.	
CICS USERID	contains the Connect:Direct CICS Interface signon ID and is eight characters long.
CICS TERMID	contains a valid terminal ID and is four characters long.
Connect:Direct NODE	contains a valid Connect:Direct node name and is 16 characters long.

Understanding the Administration Interface

In order for the user interface to perform its function, there must be a connection between it and a Data Transmission Facility. This facility can be local (that is, within the same VTAM domain) or remote (residing in another domain). This connection is accomplished using operating system subtasks that are attached in the CICS address space at Connect:Direct startup. The subtasks perform two types of functions:

- ◆ One or more worker subtasks (WSTs) are attached for each active link to a DTF. These subtasks are responsible for establishing a link to the DTF, passing commands to it, and receiving any results.
- ◆ A single controller subtask (CST) is created to monitor the work of all WSTs in the system and perform communications functions with the user interface portion of Connect:Direct.

Operating System Subtasks

Initialization of the Connect:Direct CICS Interface invokes programs that perform the following functions:

- ◆ Acquires Connect:Direct work areas, such as the signon table, the work queue areas, etc.
- ◆ Enables a task control global exit point and its associated global work area.
- ◆ Reads the configuration information from the configuration file and places that information in the global work area.
- ◆ Attaches the CST subtask.
- ◆ Checks for any DTF nodes that are to be activated at initialization and passes information to the CST task to allow it to create the appropriate WSTs and establish the link with the DTF.

Note: Connect:Direct PLTPI module is also available for interface autostart at CICS startup.

Monitor Transaction

In addition to these operating system subtasks, a CICS monitor transaction accepts input from the user interface, passes it to the CST, and routes the output to the appropriate user. This transaction also detects a loss of the CST subtask due to an error condition and performs a restart. The monitor transaction is activated at Connect:Direct startup and remains in the system for the life of the online region. Monitor activities include the following:

- ◆ Restarting the interface in case of an abnormal end
- ◆ Watching for DTF requests which are taking too long
- ◆ Forcing retry of requests still in the queue
- ◆ Forcing retry of DTF node session establishment
- ◆ Attempting to switch from ESF to primary (DTF connected) mode
- ◆ Clearing nonterminal signon table entries, at end of transaction
- ◆ Clearing signon table entries in case of abnormal ends

Interface Tasks

Although CICS runs a number of subtasks (such as journaling and VSAM handling), CICS can be regarded as a single task. All CICS transactions, which can generate Connect:Direct requests, run from this single CICS task. Connect:Direct runs as a set of separate tasks in the CICS address space. At the startup of the Connect:Direct interface, a subtask called CST (controller subtask) is attached by CICS.

Interface Startup

This isolates CICS from all the non-CICS work involved in communicating to the DTF through the DTF API. A Connect:Direct monitor transaction is also invoked as part of interface startup to support and monitor interface operation.

The CST in turn attaches other subtasks, called WSTs (worker subtasks), organized by node. The WSTs establish and manage the DTF sessions, passing commands, and receiving returned information. The CST controls the traffic between all of the Connect:Direct transactions and each WST.

Interface Subtask Management

Each WST handles requests for Connect:Direct users through the CST. One WST communicates with only one DTF, but multiple WSTs can communicate with one DTF. The DTF-connected WSTs are not associated with any particular CICS transactions. The various Connect:Direct requests generated by users are handled by any of the multiple WSTs which have been attached for a particular DTF node. The CST oversees all the WSTs and is responsible for coordinating work generated by CICS transactions.

One WST is attached per node at Connect:Direct startup, as defined in the CONFIGURATION file. Additional WSTs for a node are attached as needed, based on concurrent Connect:Direct user demand. The maximum number of WSTs (tasks) attached globally for the interface as well as per DTF node is defined with CONFIGURATION CONTROL and DTF NODE parameters. The DTF node connection limit is edited to be no larger than the number of CICS interactive applications specified in the Network Map for that node.

When an attached WST becomes inactive (has not been used for any requests) for a period of time, it is detached by CST. The inactive interval is defined with a CONFIGURATION control option. You can suppress WST activation by DTF node through a CONFIGURATION DTF node parameter.

Interface Request Management

Connect:Direct CICS Interface user requests are placed on a work queue with one queue per node. The one or more WSTs attached per node dispatch work from the queue. Maximum queue length by node is defined with a CONFIGURATION DTF node parameter which defines the number of queue entries.

When WST (session limit) is reached and additional WSTs cannot be attached to dispatch work from the queue, requests remain on the work queue until a WST becomes available. When the

number of allowed WSTs for a given node is held to a minimum, yet user activity for the given node is high, the number of queue entries should be increased.

Work queue size is also affected by the CONFIGURATION control option worker retry interval which defines the time elapsed between attempts to obtain a free WST to dispatch work present on the work queue. Connect:Direct user requests are rejected with a DTF busy message, when a DTF node queue reaches its maximum allowed number of entries.

There is a CONFIGURATION DTF node parameter to define worry time. A message is written to the log when a request to that node takes longer than the specified amount of time. There is no action taken by the Connect:Direct interface to automatically abort any requests which take too long.

Connect:Direct also has one system queue for system commands, such as SHUTDOWN, QUIESCE, etc., to dispatch system requests. The system queue size cannot be adjusted.

Interface VTAM Session

In cases where a WST/DTF VTAM session was active, but becomes inactive, the Connect:Direct interface quiesces the node. Pending requests are allowed to complete, even though they may fail. The number of WSTs for the node is reduced to one.

At this point, if the WST was in session with a local DTF, and ESF mode is allowed, the WST switches to ESF mode. An ESF MODE message is displayed to IUI users, stating that a Connect:Direct session error occurred, but ESF MODE is available. Under ESF operation, only the SUBMIT options are displayed on the PRIMARY MENU.

ESF mode operation is toggled on for the entire interface through a CONFIGURATION control parameter; ESF mode operation is toggled on for each user through a user profile parameter. When a WST fails to establish or drops a DTF session, and ESF is not allowed, then the WST will remain attached, but all user requests for the node are rejected.

The WST will periodically try to establish or re-establish (retry) a session with the associated DTF. The CONFIGURATION control parameter session retry interval defines the time lapsed between retries for session connection. An additional CONFIGURATION control parameter, the ESF session retry interval, defines the time elapsed between retries of a dummy DTF session to check if ESF mode has returned to primary mode.

Using the Administration Interface Screen

You access the Connect:Direct CICS Interface ADMINISTRATION INTERFACE screen by selecting option **I** from the PRIMARY MENU and pressing **ENTER**. The INTERFACE screen is the key to activating, monitoring, and shutting down the interface between CICS and the active DTF nodes. From the INTERFACE screen, you can manually initialize and terminate the operating system subtasks that perform the interaction with Connect:Direct. The INTERFACE screen is shown in the following figure.

```

Connect:Direct for CICS ADMINISTRATION                               10:30:37
                        INTERFACE
OPTION ==>                                     Connect:Direct for CICS VER 03
                                                REL 02
                                                MOD 00

                        A ... ACTIVATE INTERFACE
                        M ... START MONITOR
                        I ... SHUTDOWN INTERFACE (IMMEDIATE)
                        S ... SHUTDOWN INTERFACE (NORMAL)

INTERFACE STATUS          ACTIVE
PENDING REQUEST          NONE
MONITOR TASK NUMBER      25
ACTIVE TASKS             0

PF keys:  3 Exit  ENTER Refresh/Process

```

Field	Description
OPTION	This 1-character field contains the A, M, I, or S options: <ul style="list-style-type: none"> A Selecting this option, and pressing ENTER, activates the interface and automatically starts the monitor transaction. M Selecting this option, and pressing ENTER, starts the monitor transaction. This option should only be used in the event of a monitor transaction abend. I Selecting this option, and pressing ENTER, performs an immediate (hard) shutdown of the interface. All Connect:Direct user sessions are terminated, regardless of status. S Selecting this option, and pressing ENTER, performs a normal (soft) shutdown of the interface. All Connect:Direct user sessions are allowed to complete execution.
Connect:Direct CICS Interface VER	This 2-character field displays the version number of the Connect:Direct CICS Interface software.
Connect:Direct CICS Interface REL	This 2-character field displays the release number of the Connect:Direct CICS Interface software.
Connect:Direct CICS Interface MOD	This 2-character field displays the modification number of the Connect:Direct CICS Interface software.
INTERFACE STATUS	This 21-character field displays ACTIVE or INACTIVE, depending upon the state of the interface. The field contains ACTIVE if the interface is active, and will change to INACTIVE if the interface is deactivated.
PENDING REQUEST	This 18-character field displays either NONE if no activate/shutdown request is pending or the type of request being processed.
MONITOR TASK NUMBER	This 11-character field displays the number of the monitor transaction, or a message such as NOT RUNNING, if the interface is not active.
ACTIVE TASKS	This 2-character field displays the number of the active tasks.

Operational Considerations

The Connect:Direct CICS Interface can be used to sign on to multiple data transmission facility (DTFs) on either local or remote processors. To take advantage of this facility, you must consider the following:

- ◆ You must provide VTAM access to the DTF you wish to sign on to. For local DTFs (within the control of the same VTAM subsystem), you must provide IUI APPLIDs for use by both the ISPF (for MVS) and Connect:Direct IUI facilities. If the DTF facilities are located on remote processors, you must define the IUI APPLIDs for those DTFs to the local VTAM subsystem as cross-domain resources.
- ◆ For each DTF you wish to access through the IUI, you must define a DTF node record for that facility in your Connect:Direct configuration file. You can define the DTF node record by using the configuration update facilities of the CDA transaction.
- ◆ The DTF node record in your configuration file must include the DLBL of a Netmap file in order to communicate with the remote DTF. This Netmap must have the remote DTF defined as an adjacent node (through the ADJACENT.NODE NETMAP definition statement). The Network Map allows segregation of API VTAM APPLIDs such that Connect:Direct can have a pool of APPLIDs and DMBATCH can have a pool.

Note: All DTF nodes do not have to use the same Netmap file. For instance, if you wish to communicate with two DTFs with different Netmap contents, you can specify an alternate Netmap in the CICS JCL and in the DTF node record in order to communicate with the second DTF, as long as that DTF is defined as an adjacent node in the Netmap.

- ◆ All Processes to be submitted to a remote DTF facility must reside in a sublibrary in the search chain defined for your CICS.

Signing On to One DTF from Multiple Connect:Direct CICS Interface Facilities

Not only can you sign on to multiple DTF facilities from a single Connect:Direct IUI, but you can also sign on to a DTF from multiple IUI systems. If you wish to sign on in the latter fashion, please note the following for MVS users:

- ◆ If you are using the DMGSECUR macro to generate your DTF security exit and provide a value for the CICSID keyword at exit generation, all CICS systems accessing that DTF must specify the same CICSID in their signon requests as was specified in the CICSID keyword. The password for the Connect:Direct signon to a DTF will always be CICSIUI. If the CICSID keyword is not specified in the DMGSECUR macro, no checking of CICSIDs for CICS signon requests will be performed in the signon exit; however, the CICSID value with a password of CICSIUI will be passed to your security facility (if available) or to the Authorization Facility for validation. The CICSID to be used when signing on to a particular DTF is specified in the DTF NODE configuration record for that DTF.

CICSID=(whatever USERID was specified on the DTF node record screen). If you let CICSID default at installation time, the value will be CICSUSER. For example:

CICSID=CICSUSER

Reassemble the supplied security exit for the value to take effect.

- ◆ If you do not want to use the DMGSECUR macro to generate your DTF security exit, you can recognize the Connect:Direct dummy signon by checking the password, which will always be CICSUI. When a dummy signon is received from Connect:Direct, your security exit should return an Authorization Bit Mask (ABM) of binary zeros.
- ◆ To implement the CICS IUI on a base Connect:Direct that has Stage 2 security turned on, you will need to modify the supplied security exit. These exits are described in Chapter 6, *Planning and Controlling Security*. The exit to be modified depends on which security product is running on the system.

Shutting Down CICS

The Connect:Direct IUI facility provides two facilities for orderly termination of the interface, as follows:

- ◆ Termination is performed automatically by the Connect:Direct monitor transaction upon detection of a normal termination of CICS (through a CEMT PERFORM SHUTDOWN command).
- ◆ The interface can be terminated manually through the CDA transaction.

If you perform an immediate CICS shutdown (through the CEMT PERFORM SHUTDOWN IMMEDIATE command), or if CICS terminates abnormally, you will receive system OS12I SUBXXXXXX CANCELLED. MAINTASK TERMINATION from the interface for VSE. The ABENDs are generated as a result of subtask termination without detaching all the operating system subtasks created by the Connect:Direct IUI facility. In order to avoid the additional ABENDs, you must terminate the IUI facility manually, through the CDA transaction, before you issue the CEMT PERFORM SHUTDOWN IMMEDIATE command.

Restart Considerations for Task ABENDS

There are administrative options to either quiesce (soft shut, for example, allow pending requests to complete) or immediately shut down a specific node or the entire interface. Once the shutdown has been completed, another administrative option allows the restart of the interface.

In case the entire interface abends, CICS is notified. The abnormal termination is recorded in the CICS CWA. The Connect:Direct for CICS monitor transaction, when it wakes up next, detects that the interface has failed, and will try to automatically restart the interface. Users with requests to the interface when it crashed are freed by the Connect:Direct monitor, and the user is sent a message explaining the problem.

If a WST abends, then the CST is notified, if any request from a CICS user was pending, CST will fill in the return code and message for the user, informing the user that the command may have failed. If the failing WST was the only one running for a node, CST will attempt to reattach the WST.

Accounting and Logging Information

Connect:Direct accounting is accomplished by the DTF. Accounting and statistics are gathered accurately as the DTF allows the USERID to be extracted from the UICB for each Connect:Direct command entered.

CICS logging is accomplished in the background of Connect:Direct operations, but does not record all Connect:Direct events and does not duplicate any other logs. Certain of these events also display on the system console, where major events and errors are reported, such as the following:

- ◆ Interface startup
- ◆ CST attach
- ◆ WST attaches
- ◆ Node signon of dummy CICS ID
- ◆ WST session failures
- ◆ WST detaches
- ◆ CST detach
- ◆ CST termination
- ◆ Administrative commands affecting sessions and requests
- ◆ Return information for CICS users who issue a request and then abnormally exit Connect:Direct without waiting for the response

ESF Operation

ESF mode is invoked when an active DTF session fails or when session establishment fails. In order for Connect:Direct to activate ESF, the ESF option must be installed on the local DTF and the option must be enabled in the CONFIGURATION file.

The SIGNON command must also include the parameter ESF=YES. In ESF mode, a node is available for use only for SUBMIT commands which are written directly to the local DTF TCQ file. ESF SUBMIT requests can be issued only by those users who have ESF specified as a profile (signon defaults) option.

Connect:Direct SIGNON Parameters

There are four parameters in the SIGNON command explicitly to support the IUI as follows:

- ◆ TYPE=CICS parameter is added to allow a SIGNON command to be embedded in the middle of an API session. This parameter serves no other function and is invalid for a normal signon.
- ◆ TDEXIT=modname parameter is added to enable specification of the exit to receive control for temporary data set I/O. This parameter is mutually exclusive with the TMPDD, TMPDSN, UNIT, and VOLSER parameters. This exit is called for OPENS, CLOSEs, and WRITEs to the temporary data set.
- ◆ TDLIMIT=nnnnn parameter is added to restrict the number of statistics. Data is returned from the DTF for the IUI Select Statistics (SS) function. The API SIGNON (used by Connect:Direct WSTs to connect to an DTF) includes a LIMIT= PARAMETER RECORDS

returned by the DTF. The DTF truncates data sent to the WST API and appends a final record indicating that excessive output was truncated.

- ◆ NETDD=DLBL parameter is added to allow a SIGNON command to specify the DLBL of an Netmap data set which has already been allocated.

Connect:Direct CICS Interface Data Sets

The following sections detail the data sets used by DTF and Connect:Direct CICS Interface.

Connect:Direct DTF Data Sets

The following data sets, accessed in read-only mode under CICS, are used by DTF and Connect:Direct CICS Interface:

Data Set	Description
NETMAP	At least one file for each CICS region, defined as a CICS file, updated in batch, and contains all nodes available to this Connect:Direct environment; some of the information for the network nodes is also contained in the CONFIGURATION file along with additional fields that can be updated.
MESSAGE	One file for each system; defined as a CICS file; updated in batch using the message load utility; allocated and opened by CICS; VSAM KSDS; KEY = MESSAGE NUMBER inquiry under Connect:Direct to display message text.
TCQ	One Transmission Control Queue file for each node; not defined as a CICS file; holds submit (ESF) requests when DTF is down; used at the subtask level by the API only; allocated by CICS and opened by a subtask; if necessary, can be dynamically allocated and de-allocated using CICS transaction ADYN.
Trace Files	<p>Several files within the API are dedicated to system trace functions, but are not explicitly used by the Connect:Direct interface.</p> <p>However, API design allows trace data to be written when available. To capture trace data add the appropriate ddname or DLBL statements to the CICS startup JCL. The files must be present at CICS initialization.</p> <p>Note: Do not dynamically allocate trace files after CICS initializes. Each API, one per WST, will write trace data to these files when present. Examples are: NDMCMDS, which can be used to view all commands submitted to the API, and RPLERRC, which is used to write VTAM errors.</p> <p>The following example illustrates how to allocate the TRACE files:</p> <pre>// DLBL NDMCMDS, 'SYSOUT.SYS050' // ASSGN SYS050, SYSLST // DLBL RPLERRC, 'SYSOUT.SYS051' // ASSGN SYS051, SYSLST</pre> <p>Note: The SYS numbers can be any UNUSED SYS number in the CICS partition. If you change the SYS numbers, you must change on BOTH the DLBL and ASSGN statements. Also, do NOT change the SYSOUT in the FILE-ID to another name. This allows VSE/POWER to intercept the output from NDMCMDS and/or RPLERRC and route it to the VSE/POWER SYSLST for CICS. An EXTENT statement is NOT needed.</p>

Connect:Direct CICS Interface Data Sets

The following data sets, used only by Connect:Direct CICS Interface, are new for the Connect:Direct environment and are updated under CICS:

Data Set	Description
CONFIGURATION	One file for each CICS region; defined as a CICS file; updated on line through administrator functions; VSAM KSDS; contains all nodes available to this Connect:Direct environment as does NETMAP, but CONFIGURATION contains additional fields; contains system parameters that control the environment.
USER PROFILE (SIGNON DEFAULTS)	One file for each CICS region; defined as a CICS file; VSAM KSDS, key = CICS userid; updated using signon defaults function. Used to set up auto-signon to Connect:Direct.

Configuring the Connect:Direct CICS Interface Administration System

The Connect:Direct CICS Interface provides a variety of configuration options to allow you to define the resources that can be accessed using Connect:Direct, limit the scope of functions provided to users, and optimize the performance of the system. All Connect:Direct CICS Interface configuration information is in a single file (CONFIG), which is defined and initially loaded at product installation. The configuration parameters group into three categories:

- ◆ Control Record
- ◆ DTF Node Records
- ◆ Network Node Records

Each of the categories is accessed from the Connect:Direct CICS Interface ADMINISTRATION CONFIGURATION screen by selecting option C from the PRIMARY MENU. From the CONFIGURATION screen you can display and maintain the Control Record, DTF Node Records, and the Network Node Records. The sections following the CONFIGURATION screen describe each of these categories and explain how to change these configurations.

```

CONNECT:Direct for CICS ADMINISTRATION          10:25:28
                                CONFIGURATION
OPTION ==> _
                                C ... CONTROL RECORD
                                D ... DTF NODE RECORDS
                                N ... NETWORK NODE RECORDS

PF keys:  3 Exit
    
```

Entry Fields

Field	Description
OPTION	contains the C, D, or N option. This field is one character long.

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

Configuring the Control Record

The Connect:Direct CICS Interface control record contains information defining the operational characteristics of the system. This information includes flags that activate or deactivate system-wide features of Connect:Direct and parameters that govern the performance of the system.

The control record is initially loaded during the installation process; the Connect:Direct online administration facilities allow you to modify the contents of this record. Any modifications performed while Connect:Direct is active are immediately reflected in the execution environment.

You access the Connect:Direct CICS Interface ADMINISTRATION CONTROL RECORD UPDATE screen by selecting option C from the CONFIGURATION screen, and pressing **ENTER**. Only one CONTROL RECORD exists for the Connect:Direct CICS Interface, and it contains global configuration parameters. The Connect:Direct CICS Interface CONTROL RECORD UPDATE screen is shown in the following figure.

CONNECT:DIRECT FOR CICS ADMINISTRATION		11:34:35	
CONTROL RECORD UPDATE			
AUTO.SIGNON	Y	Y	OR N
SIGNON.REENTRY	Y	Y	OR N
CONNECT:Direct.EQ.CICSID	N	Y	OR N
SKIP.SIGNON.PANEL	N	Y	OR N
CICS.TRANSACTION.CODE (MONITOR)	CDM		
CICS.TRANSACTION.CODE (STARTUP)	CDS		
CICS.TRANSACTION.CODE (PRINT)	CDP		
CST.RETRY.INTERVAL	001000	HH	MMSS
SESSION.RETRY.INTERVAL	0200	MM	SS
ESF.RETRY.INTERVAL	010000	HH	MMSS
WORK.RETRY.INTERVAL	0100	MM	SS
MONITOR.INTERVAL	10	SS	
INACTIVE.INTERVAL	010000	HH	MMSS
MAX.SIGNON	0075		
MAX.TASKS	10	01	-99
MENU OPTIONS:			
CF Y	SB Y	SS Y	SP Y SD Y SN Y MD Y Y OR N
PF keys: 3	Exit	9	Update ENTER Edit CLEAR Reset

Entry Fields

Field	Description
AUTO.SIGNON	specifies whether Connect:Direct CICS Interface is to automatically sign a user on if a signon defaults record has been defined with a CICS userid matching that specified by the user at CICS signon. The signon defaults record for that CICS user must specify a valid Connect:Direct userid, password and DTF node name. If Y is specified, auto-signon will be used if the required information is available; if N is specified, auto-signon will not be used. If the AUTO.SIGNON parameter is not specified during installation, it will default to Y.
SIGNON.REENTRY	specifies whether Connect:Direct is to remember that a CICS user has previously signed on to the IUI. If this feature is enabled, a user will be able to exit Connect:Direct to perform some other CICS function and re-enter without having to sign on to Connect:Direct again. Signon reentry is not performed for a user that signs off of CICS and signs back on again between Connect:Direct sessions, and is only in effect for that user after the first signon to Connect:Direct. If Y is specified, signon reentry will be performed when appropriate; if N is specified, signon reentry will not be available. If the SIGNON.REENTRY parameter is not specified during installation, it will default to N. Note: When using this option with Y specified, a DTF shutdown with the Quiesce option will not terminate the DTF.
Connect:Direct.EQ.CICSID	specifies whether Connect:Direct will deny a signon attempt if the Connect:Direct userid does not match the CICS userid specified at CICS signon. If Y is specified, a signon will be rejected if the IDs do not match; if N is specified, no check will take place. If the Connect:Direct.EQ.CICSID parameter is not specified during installation, it will default to Y.
SKIP.SIGNON.PANEL	specifies an optional Connect:Direct signon interface that does not require users to re-enter a userid and password. Validity of this approach depends upon a secure environment existing prior to the user selecting Connect:Direct; in other words, userid/password validation by a security subsystem, such as RACF, upon original signon to the system. This option is enabled by setting SKIP.SIGNON.PANEL=YES in the control record. This can be done by entering Y for this parameter on the Control Record Update screen.
CICS.TRANSACTION.CODE (MONITOR)	This parameter specifies the 1-4-character transaction codes to be used for the Connect:Direct monitor transaction. The monitor transaction scans for pending requests from users and for completed work by Connect:Direct. If the MONITOR TRANSACTION CODE is not specified during installation, it will default to CDM. Note that if this parameter is changed, the supplied transaction definition must also be changed.

Field	Description
CICS.TRANSACTION.CODE (STARTUP)	This parameter specifies the character transaction codes to be used for the Connect:Direct start transaction. If the startup PLT is used to activate Connect:Direct at CICS initialization, this transaction is submitted to run immediately following the completion of CICS initialization processing. If the STARTUP TRANSACTION CODE parameter is not specified during installation, it will default to CDI. Note that if this parameter is changed, the supplied transaction definition must also be changed.
CICS.TRANSACTION.CODE (PRINT)	This parameter specifies the 1-4-character transaction codes to be used for the Connect:Direct print transaction. This transaction is attached to the CICS printer in response to Connect:Direct print requests. If the PRINTER TRANSACTION CODE parameter is not specified during installation, it will default to CDP. Note that if this parameter is changed, the supplied transaction definition must also be changed. Print requests are handled by writing print lines to the CICS Transient Data Area (TDA). When the data is ready for output, a transaction sends the data to the CICS printer specified in the SIGNON DEFAULTS.
CST.RETRY.INTERVAL	This parameter specifies the time interval (in hours, minutes and seconds) to be allowed to pass between attempts to restart an abended CST (controller subtask). The controller subtask is an operating system subtask responsible for monitoring the worker subtasks responsible for interaction with active DTFs defined in DTF Node Records (that is, DTFs to which Connect:Direct users can sign on). If the CSF.RETRY. INTERVAL is not specified during installation, it will default to 000500 (5 minutes).
SESSION.RETRY.INTERVAL	This parameter specifies the time interval (in minutes and seconds) to be allowed to pass between attempts to establish a VTAM session with a DTF defined in a DTF Node Record (that is, a DTF to which Connect:Direct CICS Interface users can sign on directly). If the SESSION.RETRY. INTERVAL parameter is not specified during installation, it will default to 100 (1 minute mnn).
ESF.RETRY.INTERVAL	This parameter defines the time interval (in hours, minutes and seconds) between attempts to establish a primary session with a DTF when, during a prior attempt, the DTF was not active and the node was activated in ESF mode. A node will be activated in ESF mode only if the DTF it defines supports ESF. If the ESF.RETRY.INTERVAL parameter is not specified during installation, it will default to 001500 (15 minutes).
WORK.RETRY.INTERVAL	This parameter specifies the time interval (in minutes and seconds) between the time a unit of work is submitted but cannot be placed in the work queue for a particular node and the time that unit of work is canceled. The size of the work queue for a particular node is governed by the ENTRIES IN WORK QUEUE parameter in the DTF Node Record. If the WORK.RETRY.INTERVAL parameter is not specified during installation, it will default to 0015 (15 seconds).

Field	Description
MONITOR.INTERVAL	This two-character field contains the time interval, expressed as seconds (SS), between scans for work by the monitor transaction. The monitor watches all Processes, queues, tasks, and task lengths, and regulates the flow of tasks in the system to ensure that any particular task does not seize the computer resources. When all Processes and tasks are completed or pending action by the DTF, the monitor waits for the specified interval before rescanning the work queue. If the MONITOR. INTERVAL parameter is not specified during installation, it will default to 30 (seconds).
INACTIVE. INTERVAL	This parameter specifies the amount of time (in hours, minutes and seconds) that a worker subtask (WST) is allowed to be inactive before it is detached by the controller subtask (CST). If the INACTIVE. INTERVAL parameter is not specified during installation, it will default to 003000 (30 minutes).
MAX.SIGNON	This 4-character field contains the maximum number of entries that will be built in the Connect:Direct signon table and reflects the relative size of the signon table file. If the MAX.SIGNON parameter is not specified during installation, it will default to 100 entries.
MAX.TASKS	This 2-character field contains the maximum number of simultaneous subtasks that can be attached in the CICS address space. This field should be specified as the total of the worker subtask counts for all DTF Node Records defined to Connect:Direct. Default is 2.

Menu Options

The following options contain a **Y** or **N** which enables or disables the options on the PRIMARY MENU of the general user. You can use these flags to determine the level of functionality to be provided by Connect:Direct. These options affect all users of Connect:Direct; if you wish to limit the functionality available to specific users, you must use either the authorization functions of Connect:Direct (select only the appropriate options when defining that user in the Authorization File) or the Security exit (set an Authorization Bit Mask to allow or restrict the appropriate options). All these options default to **Y**.

Option	Description
CF	This field contains the toggle to turn off the COPY FILE option on the PRIMARY MENU. The field is one character long. Y lets you copy files; N denies permission.
SB	This field contains the toggle to turn off the SUBMIT PROCESS option on the PRIMARY MENU. The field is one character long. Y permits you to submit Processes; N denies permission.
SS	This field contains the toggle to turn off the SELECT STATISTICS option on the PRIMARY MENU. The field is one character long. Y permits you to select statistics; N denies permission.
SP	This field contains the toggle to turn off the SELECT PROCESS option on the PRIMARY MENU. The field is one character long. Y permits you to select Processes; N denies permission.

Option	Description
SD	This field contains the toggle to turn off the SIGNON DEFAULTS option on the PRIMARY MENU. The field is one character with a Y or N. Y grants permission to change default signon options; N denies permission.
SN	This field contains the toggle to turn off the CHANGE SIGNON option on the PRIMARY MENU. The field is one character long with a Y or N. Y grants permission; N does not.
MD	This field contains the toggle to turn off the MESSAGE DISPLAY option on the PRIMARY MENU. The field is one character long with a Y or N. Y grants permission to use this option; N does not.

How to Change the Control Record Information

You can update the CONTROL RECORD for your Connect:Direct CICS Interface, with your changes made to the configuration parameters immediately reflected in the active system, as follows:

- ◆ Move the cursor using the tab and arrow keys to the desired field to change.
- ◆ Type in the new data.
- ◆ Press **PF9** to update the record in the Connect:Direct CICS Interface configuration file.

Configuring the DTF Node Records

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

The DTF node records contain information identifying any node eligible to be signed on to by Connect:Direct and defining the rules governing access to that node (such as ESF access and output limits). One DTF node record is required for each DTF to which Connect:Direct can sign on directly. The first of these records is defined and loaded at product installation.

DTF node information describes each DTF node available to the interface. At the startup of the interface, a subtask called CST (controller subtask) is attached by CICS. The CST in turn attaches other subtasks, called WSTs (worker subtasks). WSTs establish and manage a VTAM session with a DTF, passing commands, and receiving returned information. One WST communicates with only one DTF, but multiple WSTs can communicate with one DTF.

How to Update the DTF Node Records Information

You can update the DTF NODE RECORDS of all the users of the Connect:Direct system. Updating can be performed by adding, deleting, reading, and editing the DTF node records, as follows:

Adding a DTF Node Record

To add a DTF record, perform the following steps:

1. Type in the NODE NAME.
2. Type in the NETMAP DDNAME.
3. Type in the DUMMY ID FOR DTF SIGNON.
4. Type in the **Y** or **N** for SUPPRESS CONNECTION AT STARTUP.
5. Type in the **Y** or **N** for ESF SIGNON ALLOWED.
6. Type in the number of MAXIMUM WORKER SUBTASKS.
7. Type in the number of ENTRIES IN WORK QUEUE.
8. Type in the number of bytes for OUTPUT RECORD LIMIT.
9. Type in the time (MMSS) for SLOW RESPONSE NOTIFICATION.
10. Press **PF5** to add the configuration file.

Deleting a DTF Node Record

To delete a DTF node record, perform the following tasks:

- ◆ Display the DTF node record you wish to delete, either by using **PF7** and **PF8** to scroll through the list, or by typing in the node name, and pressing **ENTER**.
- ◆ Press **PF6**.

Reading a DTF Node Record

To read a DTF node record, use either one of the two methods to read a DTF node record, as follows:

- ◆ Type in the DTF Node Record you want to read.
- ◆ Press **ENTER**.

Editing a DTF Node Record

To edit a DTF node record, perform the following tasks:

- ◆ Display the DTF node record you wish to update, either by using **PF7** and **PF8** to scroll through the list, or by typing in the node name, and pressing **ENTER**.
- ◆ Move the cursor to the desired fields by using the tab and arrow keys.
- ◆ Press **PF9** to update the record in the configuration file.

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

DTF Node Records Screen

The Connect:Direct CICS Interface Administration DTF Node Records screen is accessed by selecting option **C** on the PRIMARY MENU, then by selecting option **D** on the

CONFIGURATION screen, and pressing **ENTER**. One DTF node record exists for each DTF node which users can sign on to directly. The DTF Node Records screen is shown in the following figure.

CONNECT:DIRECT FOR CICS ADMINISTRATION		11:45:48
DTF NODE RECORDS		
DTF NODE NAME	"nodename"	
NETMAP DDNAME	"ddname"	
DUMMY ID FOR DTF SIGNON	"id"	
SUPPRESS CONNECTION AT STARTUP	Y	Y OR N
ESF SIGNON ALLOWED	Y	Y OR N
MAXIMUM WORKER SUBTASKS	03	
ENTRIES IN WORK QUEUE	050	
OUTPUT RECORD LIMIT	05000	
SLOW RESPONSE NOTIFICATION	0030	MMSS
PF keys: 3 Exit 5 Add 6 Delete 7 Prev 8 Next 9 Update		
ENTER Read/Edit CLEAR Reset		

Entry Fields

Field	Description
DTF NODE NAME	This 16-character field contains the name of a data transmission facility node.
NETMAP DDNAME	(For MVS) specifies the ddname of the NETMAP file to be used when initiating a signon to this node. For VSE, this parameter specifies the DLBL of the NETMAP file to be used when initiating a signon to this node. Note: The NETMAP DLBL name appears in the File Control Table and in the CICS startup JCL. You must have a NETMAP file for each node and you must have the DTF node record for each node you want to sign on to.
DUMMY ID FOR DTF SIGNON	specifies the USERID to be used to initially establish the VTAM session with the DTF. Multiple CICS userids are required if a DTF can have multiple Connect:Direct systems signed on concurrently. For MVS, if you are using the DMGSECUR macro to define your Connect:Direct security exit, this parameter must match the CICS parameter coded for that macro. The default is CICSID.
SUPPRESS CONNECTION AT STARTUP	is a 1-character field containing the toggle for startup connections. If the field contains Y, the connection between the specified DTF node and the Connect:Direct system is suppressed when the software is started. If the field contains N, then the connection is made. During installation, if this parameter is not specified, it defaults to Y.
ESF SIGNON ALLOWED	specifies whether the extended submit facility is to be supported for this node by Connect:Direct CICS Interface. The DTF identified in this record must support ESF in order to sign on in ESF mode using Connect:Direct. During installation, if this parameter is not specified, it defaults to Y.
MAXIMUM WORKER SUBTASKS	specifies the maximum number of WSTs (worker subtasks) to be used for this node. There should be one WST per parallel session defined for this node. The total number of worker subtasks defined for all DTF NODE records in the system should not exceed the MAX TASKS value in the control record. During installation, if this parameter is not specified, it defaults to 2.

Field	Description
ENTRIES IN WORK QUEUE	specifies the maximum number of actual requests to be allowed on the pending work queue for this node. An excessive value here could result in an inordinate response time for terminal users. During installation, if this parameter is not specified, it defaults to a value two times the MAXIMUM WORKER SUBTASKS specification.
OUTPUT RECORD LIMIT	specifies the upper limit on the number of lines of output that will be accepted in response to a SELECT PROCESS or SELECT STATISTICS command. Output from these commands is stored in CICS temporary storage until viewed or explicitly deleted. During installation, if this parameter is not specified, it defaults to a limit of 100 eighty-byte records.
SLOW RESPONSE NOTIFICATION	is a 4-character field containing the time interval, expressed as minutes and seconds (MMSS), after which Connect:Direct will notify you of potential problems with slow response. During installation, if this parameter is not specified, it defaults to 0200 (2 minutes).

Configuring Network Node Records

To facilitate the task of copying files from one node to another, Connect:Direct provides a means of predefining often-used nodes, relieving the user of the need to know specific information about nodes to be accessed. Nodes defined in Network Node Records will appear in a numbered list in the COPY FILE BETWEEN NODES screen; users can select nodes to be sent to or received from by number and provide only that information relating to the file to be sent or received.

How to Update the Network Node Records Screen

You can key in a node name and press **ENTER** to get information for that node, and you can update the network node records of all the users of the Connect:Direct CICS Interface. Updating can be performed by adding, deleting, and editing the network node records.

Adding a Network Node Record

To add a network node record, perform the following steps:

1. Type in the NETWORK NODE NAME
2. Type in the NETWORK NODE DESCRIPTION
3. Type in NODE TYPE
 - ◆ 1=MVS
 - ◆ 2=VM
 - ◆ 3=VSE
 - ◆ 4=VMS
 - ◆ 5=Tandem

- ◆ 6=MS-DOS
 - ◆ 7= OS/2
 - ◆ 8=OS/400
 - ◆ 9=UNIX
 - ◆ 10=NETWARE
 - ◆ 11=Windows NT
4. Press **PF5** to add the configuration file.

Deleting a Network Node Record

To delete a network node record, perform the following tasks:

1. Display the network node record you wish to delete, either by using **PF7** and **PF8** to scroll through the list, or by typing in the node name, and pressing **ENTER**.
2. Press **PF6**.

Reading a Network Node Record

To read a network node record, use either one of the two methods, as follows:

- ◆ Method 1 - Press **PF7** and **PF8** to scroll backward and forward, respectively, to the desired NETWORK NODE NAME to read.
- ◆ Method 2 - Type in the desired NETWORK NODE NAME for reading. Press **ENTER**.

Editing a Network Node Record

To edit a network node record, perform the following tasks:

1. Display the network node record you wish to update, either by using **PF7** and **PF8** to scroll through the list, or by typing in the node name, and pressing **ENTER**.
2. Move the cursor to the desired fields by using the tab and arrow keys.
3. Press **PF9** to update the record in the configuration file.

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

Network Node Records Screen

The Connect:Direct CICS Interface Administration Network Node Records screen is accessed by selecting option **C** from the PRIMARY MENU, and then selecting option **N** from the CONFIGURATION screen, and pressing **ENTER**. The screen fields are scrollable, and allow you to view the contents of the network node records.

The Connect:Direct CICS Interface Network Node Records screen is shown in the following figure.

```

CONNECT:Direct for CICS ADMINISTRATION          14:44:36
      NETWORK NODE RECORDS

NETWORK NODE NAME      _____

NETWORK NODE DESCRIPTION _____

NODE TYPE              ___  1=MVS  2=VM  3=VSE  4=VMS  5=TANDEM  6=MS-DOS
                          7=OS/2 8=OS/400 9=UNIX 10=NETWARE 11=WINDOWS NT

PF keys:  3 Exit   5 Add   6 Delete  7 Prev  8 Next  9 Update
          Enter Read/Edit Clear Reset
    
```

Entry Fields

Field	Description
NETWORK NODE NAME	This 16-character field contains the name of the node.
NETWORK NODE DESCRIPTION	This 30-character field contains a description of the node.
NODE TYPE	This one-character field contains the environment number. Currently, valid environment numbers are as follows: 1 for MVS, 2 for VM, 3 for VSE/ESA, 4 for VMS, 5 for Tandem, 6 for MS-DOS, 7 for OS/2, and 8 for OS/400, 9 for UNIX, 10 for NETWARE, and 11 for Windows NT.

Monitoring Status with the Connect:Direct CICS Interface Administration System

With the Connect:Direct CICS Interface, you can monitor and control the node status, the user status, the work queue, and the interface status.

Node Status and Control

The Connect:Direct CICS Interface Administration Node Status screen is accessed by selecting option **N** from the PRIMARY MENU and pressing **ENTER**. The Node Status screen provides you with the ability to check the status of all DTF nodes eligible to be accessed by Connect:Direct, and you can selectively activate, deactivate, and view pending work for all DTF nodes. The Node Status screen is shown in the following figure.

CONNECT:Direct for CICS ADMINISTRATION								10:30:37	
NODE STATUS									
DTF NODE NAME	CICS STATUS	ADMIN REQUEST	SESS TYPE	SESSION DATE/TIME	OR SESSION MSGID	MAX TASKS	CURR TASKS	CURR WORK	
_ NODE1	INACT		NONE			3	0	0	
_ NODE2	ACTIVE		PRIM	06/19/1997 09:57:18		2	1	0	
Line commands: A Activate (start first task) I Shut immediate									
W Work queue display S Shut normal									
PF keys: 3 Exit ENTER Refresh/Line cmd									

Line Commands

Option	Description
A	Selecting this option in the field to the left of the DTF node name, and pressing ENTER, activates the node.

Option	Description
W	Selecting this option, and pressing ENTER, goes to the WORK QUEUE screen, if work for this node is being processed.
I	Selecting this option, and pressing ENTER, performs an immediate (hard) shutdown of the node. All processes for this node are forced to stop, regardless of status.
S	Selecting this option, and pressing ENTER , performs a normal (soft) shutdown of the node. All processes for this node are allowed to complete execution.

System Fields

Option	Description
DTF NODE NAME	This 16-character field contains the DTF node name as entered in the DTF Node Record.
CICS STATUS	This 6-character field contains the status of the node, as follows: <ul style="list-style-type: none"> ACTIVE The node has been activated, either by Connect:Direct initialization or manually. INACT The node is not active.
ADMIN REQUEST	This 8-character field contains the type of administrative request affecting the status of the node, as follows: <ul style="list-style-type: none"> ACTIVATE The node is currently being activated. SHUTNORM a normal shutdown of the node has been requested, and the node is quiescing. SHUTIMM an immediate shutdown of the node has been requested, and the link with that DTF is being terminated.
SESS TYPE	This four-character field contains the type of session held with this DTF node, as follows: <ul style="list-style-type: none"> NONE no session currently exists with this DTF node. PRIM Connect:Direct is currently in session with the DTF node. ESF The DTF is not active, but supports the Extended Submit Facility. Connect:Direct is accepting requests allowed in ESF mode.
SESSION DATE/TIME OR SESSION MSGID	This 17-character field contains either the date and time that the connection to this DTF was activated, or the MSGID of the last message issued for this DTF node during activation or deactivation.
MAX TASKS	This 5-character field contains the maximum number (high water mark) of subtasks that have been attached to process requests directed at this node.
CURR TASKS	This 5-character field contains the current number of subtasks attached to process requests directed at this node.

Option	Description
CURR WORK	This 5-character field contains the number of subtasks currently processing requests directed at this node.

How to Control Accessible Nodes

The Connect:Direct user interface provides the ability to log on to any DTF in your network and perform Connect:Direct operations using that node as your process PNODE. DTFs to be accessed must be identified in DTF Node Records in the configuration file along with the configuration parameters to be used when communicating with that node.

You can monitor the NODE STATUS of all the nodes associated with Connect:Direct. Once you are at the PRIMARY MENU, select **N** to go to the NODE STATUS screen, and then perform any or all of the following tasks:

- ◆ Press **PF7** and **PF8** to scroll backward and forward, respectively, through the list of DTF nodes connected to the Connect:Direct session.
- ◆ Move the cursor using the tab and arrow keys to the field to the left of the DTF NODE NAME.
- ◆ Type in option **A**, and press **ENTER**. Activating a node in this way connects the DTF node to the CICS session.
- ◆ Type in option **W**, and press **ENTER**. If tasks are currently being performed for the active DTF node, you go to the WORK QUEUE display.
- ◆ Type in option **I**, and press **ENTER**. Deactivating a node in this way disconnects the DTF node from the CICS session immediately.
- ◆ Type in option **S**, and press **ENTER**. Shutting down the node in this way allows the node to complete the execution of all subtasks.

Viewing the Work Queue

The Connect:Direct CICS Interface ADMINISTRATION WORK QUEUE screen is accessed by selecting option **W** from the NODE STATUS screen, and pressing **ENTER**. The WORK QUEUE screen fields contain the data describing the tasks in the work queue for each CICS user on a Connect:Direct DTF node. The Connect:Direct CICS Interface WORK QUEUE screen is shown in the following figure.

CONNECT:Direct for CICS ADMINISTRATION							10:30:37
WORK QUEUE - NODE "nodename"							Page 01 of 01
CICS ID	USERID	CICS TERM	LAST TASK#	CURR CMD	REQUEST DATE/TIME	TD CTR	WORK TASK
-----	-----	----	-----	---	-----	-----	----
ID1	ID1	M064	00271	SB	06/21/95 11:47:13	00282	0346
PF keys:	3 Exit	7 Bwd	8 Fwd	12 Node	USER STATUS	ENTER	Refresh

System Fields

Field	Description
NODE	This 16-character field contains the node name associated with the work queue data.
Page XX of YY	This 13-character field contains the number of the current page XX out of a total YY pages for the work queue list.
CICS ID	This 8-character field contains the CICS userid of the user submitting the work.
USERID	This 8-character field contains the Connect:Direct CICS Interface USERID of the user submitting the work.
CICS TERM	This 4-character field contains the terminal ID from which the work was submitted.
LAST TASK#	This 5-character field contains the transaction number of the task submitting this request.
CURR CMD	This 2-character field contains the representation of the current command executed by the task (such as CF, SB, and SS).
REQUEST DATE/TIME	This 17-character field contains the date and time the work was submitted.
TD CTR	This 5-character field contains the number of bytes (counted by the transient data counter) indicating how much data was written by the exit module for a transaction.
WORK TASK	This 4-character field contains the number of the work task.

How to View the Work Queue Display

The Connect:Direct work queue is defined for each node to dispatch Connect:Direct requests and responses. The DTF node configuration file contains information required by the Connect:Direct CICS Interface to manage the DTF VTAM or ESF sessions. Each WST uses the standard Connect:Direct API to manage the DTF or ESF session.

You can view the work queue for any node associated with Connect:Direct. Once you are at the PRIMARY MENU, select **N**, and press **ENTER**, to go to the NODE STATUS screen, and then select option **W** to the left of the desired node, and press **ENTER** again.

Once you are at the WORK QUEUE screen, perform any or all of the following tasks:

- ◆ Scroll backward and forward, respectively, using **PF7** and **PF8**, to view the entries.
- ◆ Press **PF12** to view to the USER STATUS screen, and then press **PF3** to get back to the WORK QUEUE screen.

Viewing User Status

The Connect:Direct CICS Interface ADMINISTRATION USER STATUS screen is accessed from the PRIMARY MENU by selecting option **U**, and pressing **ENTER**. The USER STATUS screen

shows you data on any signed-on user. The USER STATUS screen is shown in the following figure.

CONNECT:Direct for CICS ADMINISTRATION							10:30:37	
USER STATUS								
CICS ID	CICS TERM	SESS TYPE	USERID	DTF NODE NAME	STATUS	LAST TASK #	MSGID	
_	MASTER	M064	PRIM	ID1	"dtfnodename"	CICS	45	SAFA000I
Line commands:		F	Free user	C	Free user and cancel user's subtask			
PF keys:		3	Exit	10	Left	11	Right	ENTER Refresh/Process

Options

You would normally select these options in the underscore field to the left of the CICS USERID, when there is a user security violation or resource allocation abuse.

Option	Description
F	Selecting this option frees any user from use of the Connect:Direct system.
C	Selecting this option frees any user from use of the Connect:Direct system, and cancels that user's subtask.
T	Selecting this option frees any user from use of the Connect:Direct system, and terminates that user's signon.

System Fields

Field	Description
CICS ID	This 8-character field contains the CICS USERID of all currently signed-on users.
CICS TERM	This 4-character field contains the CICS terminal ID of all currently signed-on users.
SESS TYPE	This 4-character field contains the session type (PRIM or NONE).
USERID	This 8-character field contains the Connect:Direct USERID of all currently signed-on users.
DTF NODE NAME	This 16-character field contains the DTF node name the user is signed on to.

Field	Description
STATUS	This 7-character field contains the STATUS of a user. The following could also be in this field: <ul style="list-style-type: none"> ◆ QUEUED ◆ SIGNON ◆ SIGNOFF ◆ RUNNING ◆ CREATE ◆ DELETE ◆ CICS ◆ INPUT
LAST TASK #	This field contains the last task number of a user process.
MSGID	This field contains the message ID of the last message for a given user. Note: If you press PF11 to scroll right on the screen, the following fields will appear.
TD CTR	This 5-character field contains the number of bytes (counted by the transient data counter) indicating how much data was written by the exit module for a transaction.
LAST SIGNON	This field contains the time of the last signon.
LAST REQUEST	This field contains the time of the last request.

Monitoring User Status

In the course of online system availability, it is occasionally necessary to determine what users are accessing a given resource, what actions they are performing using that resource, and what must be done to resolve problems encountered by these users. To address these needs, Connect:Direct provides the user status function, allowing you to view information about users of the system and, if necessary, intervene to resolve error situations. You can either view all users of the system, a single user (optionally qualified by CICS userid or terminal ID) or a group of users (optionally qualified by DTF node).

Once you are at the PRIMARY MENU, perform the following tasks, using either one of the two methods.

Method 1

- ◆ From the PRIMARY MENU, select U, and the cursor will move to the CICS USERID field.
- ◆ Type in any combination of CICS USERID, CICS TERMID, and Connect:Direct NODE, and press RETURN to move to the next field.
- ◆ Press **ENTER** to go to the USER STATUS screen.

Note: The combination typed in restricts what data is displayed on the USER STATUS screen. For example, if you typed in only the Connect:Direct NODE, the USER STATUS screen shows all CICS USERIDs currently active on only that particular DTF node.

- ◆ On the USER STATUS screen, you can scroll up, down, left, and right to view your status data with **PF7**, **PF8**, **PF10**, and **PF11**, respectively. You can also type in any of the F, C, and T line commands as desired, and press **ENTER**.

Method 2

- ◆ From the PRIMARY MENU, select **U**, and press **ENTER** to go to the USER STATUS screen.
- ◆ On the USER STATUS screen, you can control user status by using the tab and arrow keys to move the cursor to the field in front of the CICS ID field.

Note: Usually, you will use these line commands only if the user is engaged in unauthorized activity, stuck in a continuous loop, or has created a task that is seizing computer CPU time, storage, or I/O resources.

- ◆ Type in any of the **F**, **C**, and **T** line commands as desired, and press **ENTER**.

Using Connect:Direct CICS Interface Signon Defaults

The Connect:Direct CICS Interface ADMINISTRATION SIGNON DEFAULTS screen is accessed by selecting option **S** from the PRIMARY MENU, and pressing **ENTER**. The Connect:Direct CICS Interface SIGNON DEFAULTS screen is shown in the following figure.

```
CONNECT:Direct for CICS ADMINISTRATION          14:59:46
                SIGNON DEFAULTS

CICS USERID          ==> _____

**CONNECT:Direct**
USERID ==> _____
PASSWORD==> _____

DEFAULT NODE          ==> _____
ESF MODE ALLOWED      ==> _  Y OR N
UPPER CASE PRINT      ==> _  Y OR N
CICS PRINTER          ==> _____
PNODE ACCT DATA      ==> _____
SNODE ACCT DATA      ==> _____

Do you want all commands for this session to be CASE sensitive? ==> NO_

PF keys:  3 Exit   5 Add   6 Delete  7 Prev   8 Next   9 Update
          Enter Read/Edit  Clear Reset
```

Entry Fields

Field	Description
CICS USERID	This field contains an acceptable USERID for the CICS signon, and is eight characters long.
Connect:Direct USERID	This 64-character field contains an acceptable USERID for the Connect:Direct CICS Interface signon .
Connect:Direct PASSWORD	The 64-character field contains the valid PASSWORD for the userid specified for Connect:Direct USERID.
DEFAULT NODE	This 16-character field contains the name of the default Connect:Direct node. The user will automatically be signed on to this node if it is active, or denied access if it is not active.
ESF MODE ALLOWED	This 1-character field contains the permission for use of the extended submit facility (ESF). If the field contains Y, permission is allowed; if the field contains N, permission is denied.
UPPER CASE PRINT	This 1-character field contains the switch for upper case printing. If the field contains Y, all printed output from Connect:Direct CICS Interface will be in upper case. If the field contains N, the printed output is in upper and lower case.
CICS PRINTER	This 4-character field contains the designation for the Connect:Direct CICS Interface printer to be used when servicing print requests for this user.
PNODE ACCT DATA	This 50-character field contains the primary node accounting data for allocation of budget to CPU time, I/O, and other computer resources.
SNODE ACCT DATA	This 50-character field contains the secondary node accounting data for allocation of budget to users for CPU, I/O, and computer resource time.
CASE Sensitive	<p>Following this question, indicate whether you want to allow mixed case input. This option is available as a session default, and you can specify the option during SIGNON. You can override the specified default on commands that apply to USERID, PASSWORD, DATA SET NAME. When commands are submitted and YES is specified, Connect:Direct includes the CASE=YES parameter with your command.</p> <p>Note: CICS only interprets mixed case data if your terminal is defined to accept it. The CICS TCT TYPE definition must be defined with UCTRAN=NO for mixed case data to be input to Connect:Direct from a CICS terminal.</p>

System Fields

Field	Description
Connect:Direct PASSWORD MESSAGE	This 44-character field indicates whether a password is present or absent in the user's signon defaults record, and occurs to the right of the Connect:Direct PASSWORD field.

Updating the Signon Defaults Screen

To speed signon and define certain characteristics of a user, Connect:Direct provides a means to predefine a user's signon information. Information such as the Connect:Direct userid and password to be used for a particular CICS userid can be placed in the Connect:Direct user profile data set. This information is not required for all users; however, it can facilitate use of the product.

When a user enters the Connect:Direct transaction, the CICS userid is retrieved and used to read the user profile data set. If a record is found, the information in that record is used to control some of the operations that can be performed by that user. If a Connect:Direct userid and password are provided in the signon defaults record, Connect:Direct CICS Interface will automatically sign that user on; otherwise, the user will be prompted for a userid and password before being allowed access to Connect:Direct. After gaining access to Connect:Direct, a user can change his particular signon defaults using the SD function.

You can update the Signon Defaults records of any users of the Connect:Direct CICS Interface system. Updating can be performed by adding, deleting, reading, and editing the signon records.

Adding a User Sign-On Record

To add a user sign-on record, perform the following tasks:

1. Type in the CICS USERID (required).
2. Type in the Connect:Direct USERID (required).
3. Type in the Connect:Direct PASSWORD (required).
4. Type in the DEFAULT NODE (required).
5. Type **Y** or **N** for ESF MODE ALLOWED (optional, default is **Y**).
6. Type in the **Y** or **N** for UPPER-CASE PRINT (optional, default is **N**).
7. Type the designation for the CICS printer (optional, default is CDP).
8. Type in the PNODE ACCT DATA (optional, no default).
9. Type in the SNODE ACCT DATA (optional, no default).
10. Press **PF5** to add the information to the Connect:Direct CICS Interface user profile data set.

Deleting a User Sign-On Record

To delete a user sign-on record, you can display the record to be deleted, as follows:

1. Type in the CICS USERID you wish to delete, or press **PF7** and **PF8** to scroll through the file until the USERID appears.
2. Press **PF6** to delete the signon defaults record.

Reading a User Sign-On Record

To read a user sign-on record, you can display the record to be read, by typing the CICS USERID, or press **PF7** and **PF8** to scroll through the file until the desired USERID appears.

Editing a User Sign-On Record

1. To edit a user sign-on record, you can display the record to be read, as follows:
2. Type in the CICS USERID, or press **PF7** and **PF8** to scroll through the file until the desired USERID appears.
3. Move the cursor with the tab and arrow keys to the desired fields.
4. Type in the new data.
5. Press **PF9** to update the signon defaults record.

Using the Connect:Direct CICS Interface Application Program Interface

In addition to the Connect:Direct CICS Interface, a facility is provided that allows you to issue standard Connect:Direct commands from a CICS application program. An example of the use of this facility is provided in member DMQ247 of the Connect:Direct CICS Interface source library; type S. This program allows the user to enter Connect:Direct commands on a screen and view the resulting return code, message number, message text, and Process number assigned to your Process. In addition, the name of the CICS temporary storage (TS) queue where the results of your command are stored (if any), the count of items in the queue, and the maximum record length in the queue are also displayed. The components required to use DMQ247 are as follows:

- ◆ DMQ247 - the program source, written in assembly language
- ◆ DMQM98 - the BMS map used by DMQ247 (DMQM98 screen is shown in the following figure)
- ◆ DMQBMSTB - a parsing macro used by DMQ247
- ◆ DMQCA012 - a command-level COMMAREA passed to the Connect:Direct CICS Interface

```

CONNECT:Direct for CICS API DRIVER                                10:30:37

COMMAND _____
COMMAND RC
COMMAND MSG ID
COMMAND MSG
PROCESS NUMBER
TD EXIT COUNT
TS QUEUE NAME
TS MAX LRECL
PF keys:   3 Exit   5 Signon   6 Command   7 Signoff

```

Driver Fields

Field	Description
COMMAND	This 62-character field contains your API command.
COMMAND RC	This 4-character field contains the return code from your API Process.
COMMAND MSG ID	This 8-character field contains the identification number of the message associated with your API Process.
COMMAND MSG	This 64-character field contains the text of the message.
PROCESS NUMBER	This 6-character field contains the Process number assigned by the system to your API Process.
TD EXIT COUNT	This 6-character field contains the number of bytes indicating how much data was written by the exit module for your API Process.
TS QUEUE NAME	This 8-character field contains the name of the TS queue used during your API request.
TS MAX LRECL	This 4-character field contains the maximum logical record length in bytes of the TS queue.

Connect:Direct CICS Interface API Sample Program

To use the sample program, first change DMQ247 to issue a valid SIGNON command for your environment. (The SIGNON command is defined near the end of the source module.) You must then assemble DMQM98 and DMQM247 using the procedures for your system, place the load modules in your CICS application library, and define these to your CICS system. Any transaction code can be used to invoke DMQ247. Processing for DMQ247 proceeds as follows:

- ◆ The program checks for the presence of a commarea in the Exec Interface Block (EIB). If none is present, or if the user pressed the **CLEAR** key, the DMQM98 map is sent and a RETURN TRANSID is performed to re-invoke the transaction when the **ENTER** key is pressed.
- ◆ If the **PF3** or **PF15** key was pressed, the program is terminated.
- ◆ The DMQM98 map is received. If **PF5** or **PF17** was pressed, a Connect:Direct SIGNON request is generated and the results of the command are presented in the map.
- ◆ Following a successful signon (the return code on DMQM98 after SIGNON is zero), the user can enter a valid Connect:Direct command on the line provided in the DMQM98 map and press **PF6** or **PF18** to send the command to the Connect:Direct DTF. Command results will be displayed when returned from the DTF.
- ◆ To signoff from the DTF, press **PF7** or **PF19**.

When a command is entered through the DMQM98 screen, its length is determined and the address of the length and command are placed in the DMQCA012 commarea at label Q012CMDA. Program DMQ012 is then invoked with an EXEC CICS LINK command. When control returns to DMQ247, the DMQCA012 commarea will contain the results of the command.

Note: The DMQ247 does not actually display the results of the issued command that are stored in CICS temporary storage. These records can be retrieved programmatically or can be viewed using the CICS CEBR transaction.

The techniques shown in the DMQ247 sample program can be used to issue any valid Connect:Direct command. Results of commands such as SELECT PROCESS and SELECT STATISTICS will be written to CICS temporary storage; other commands may produce no output.

Connect:Direct CICS Interface Administration Messages

This appendix contains messages which occur during the Connect:Direct CICS Interface administrative operations, and are arranged alphabetically.

Message	Explanation
Activate rejected, node is not inactive - XXXXXXXXXXXXXXXXXXXX	XXXXXXXXXXXXXXXXXXXX represents a nodename. This message is given if option A is selected in the field beside an active node.
All values reset from config file	This message is displayed if you press CLEAR.
All values reset from signon defaults file	This message is displayed if you press CLEAR to refresh the screen data.
CICS USERID required	This message is displayed if you pressed ENTER with a blank screen present or with no CICS USERID entry present.
Control record successfully updated	This message is given after an edit session, and you press PF9.
DTF NODE NAME required	This message is given if you pressed ENTER, or PF5, or PF6, with a blank screen.
End of file	This message is given when you scroll forward to the bottom of the file through repeated use of the PF8 key.
End of file; values read from config file	This message is displayed when you scroll forward to the end of file, and then press PF8.
End of file; values read from signon defaults file	This message is displayed if you press PF8 repeatedly, and attempt to access data beyond the end of file.
First page	This message appears if you repeatedly press PF7, and attempt to access data before the beginning of the file.
Immediate shutdown rejected; interface is shut.	This message is displayed if option I is already selected, and the interface is already inactive.
Immediate shutdown started	This message is given if option I is selected. The INTERFACE STATUS field changes to INACTIVE.

Message	Explanation
Interface has been started	This message is given if you select option A, and press ENTER.
Interface is already active	This message is displayed if option A is selected, and the interface is already active or in the process of starting.
Interface must be active to start monitor.	This message is given if option M is selected, and the interface is not active. Option A must be selected before option M.
Last page	This message appears if you repeatedly press PF8, and attempt to access data beyond the end of the file.
Left page	This message is given if you repeatedly press PF10, and attempt to access data past the left edge of the screen.
Line command invalid	This message appears if you select a different line command other than those defined, and press ENTER.
Monitor has been started.	This message is displayed if option M is selected.
Monitor is already running.	This message is displayed if option M is already selected, and option M is selected again.
NETWORK NODE NAME required	This message is displayed if you press ENTER, or PF5, or PF6, with a blank screen present, or with the NETWORK NODE NAME field entry blank.
Network node successfully added - XXXXXXXXXXXXXXXXXXXX	XXXXXXXXXXXXXXXXXXXX represents a nodename. This message is displayed when the DTF NODE RECORDS screen has your correct data entered in the fields, and you press PF5.
Network node successfully deleted - XXXXXXXXXXXXXXXXXXXX	XXXXXXXXXXXXXXXXXXXX represents a nodename. This message is displayed when the DTF NODE RECORDS screen has your correct data entered in the fields, and you press PF6.
No active work queue entries for node XXXXXXXXXXXXXXXXXXXX	XXXXXXXXXXXXXXXXXXXX represents a nodename. This message is given if option W is selected, and there are no active subtasks in the work queue.
Node activation started, node XXXXXXXXXXXXXXXXXXXX	XXXXXXXXXXXXXXXXXXXX represents a nodename. This message is displayed if option A is selected in the field beside an inactive node.
Node has been restarted - XXXXXXXXXXXXXXXXXXXX	XXXXXXXXXXXXXXXXXXXX represents a nodename. This message is displayed in the STATUS ALERT MESSAGE LINE when the NETWORK NODE RECORDS have been updated to activate a node.
NODE invalid	This message is displayed if the node name you typed in is not authorized for sign-on.
NODE required	This message is displayed if you pressed ENTER on the blank screen, without typing in a node name.
NODE TYPE invalid	This message is given if the DTF NODE NAME contains garbage or an unacceptable naming convention.

Message	Explanation
No password is currently on file	This message is displayed in the Connect:Direct PASSWORD field if you typed in an acceptable CICS USERID and Connect:Direct USERID, and an unacceptable or blank Connect:Direct PASSWORD, and then pressed ENTER.
Normal shutdown rejected; interface is inactive.	This message is given if option S is already selected, and option S is selected again.
Normal shutdown started	This message is given if option S is selected, and ENTER is pressed. If ENTER is pressed again, the MONITOR TASK NUMBER field changes to NOT RUNNING.
No signed-on users	This message is given if the administrative user selected line command T from the USER STATUS, and pressed ENTER, thereby terminating the administrative user session. This message is also displayed if you enter the U option from the primary menu and there are no signed-on users of Connect:Direct.
No users meeting selection criteria	This message is displayed if a CICS USERID was entered in the CICS USERID field that did not match those users logged on. First, check the USER STATUS screen to see who is logged on for a match.
OPTION invalid	This message is given if you enter any OPTION other than a defined option in the OPTION field.
Password is on file, but not displayed	This message is displayed if you enter a Connect:Direct PASSWORD that is already in the SIGNON DEFAULT record.
Past end of file - 'Prev' not available	This message is given when you scroll backward to the top of the file, and then press PF7.
PF key invalid	This message is given if the user pressed a PF key other than those defined on the screen.
Record cannot be deleted, it is not on file	This message is given if you type in the DTF NODE NAME, and then press PF6. There is no match between the DTF NODE NAME and those specified in the configuration file, and therefore the record cannot be deleted.
Record cannot be updated, it is not on file	This message is given if you type in the DTF NODE NAME, and then press PF9 . There is no match between the DTF NODE NAME and those specified in the configuration file, and therefore the record cannot be updated.
Record not found; hit any key when ready.	This message is displayed if you type in the DTF NODE NAME, and you press ENTER, but there is no match in the DTF NODE RECORDS.
Right page	This message is displayed if you repeatedly press PF11, and attempt to access data past the right edge of the screen.
SAFA000I - Connect:Direct signon process completed.	This message appears if you signed on correctly.

Message	Explanation
SCCS007I - DTF node not active to Connect:Direct CICS Interface.	This message is given if the node name you typed in is not active in the Connect:Direct system. Check the NODE STATUS screen.
SCCS018I - Request could not be assigned to a subtask.	This message is displayed if the user typed in incorrect data or unknown data. The system could not place the signon request in the work queue assigned to the DTF node for the length of time required to complete the signon process. Maximum users may have exceeded, or incorrect entries in the SIGNON DEFAULTS record may have caused the failure.
Shut immediate started, node XXXXXXXXXXXXXXXXXXXX	XXXXXXXXXXXXXXXXXXXX represents a nodename. This message is given if option I is selected, and ENTER is pressed on an active node.
Shut normal started, node XXXXXXXXXXXXXXXXXXXX	This message is displayed if option N is selected, and ENTER is pressed, on an active node. The STATUS field changes to INACT, and the REQUEST field changes to SHUTNORM.
Shut rejected; node already inactive - XXXXXXXXXXXXXXXXXXXX	XXXXXXXXXXXXXXXXXXXX represents a nodename. This message is given if option S is selected, and ENTER is pressed, on an inactive node.
Signon defaults successfully updated - USERID	This message is given if you press PF9 alone, or PF9 after an edit session.
Start of file	This message is displayed if you scroll back to the top of the file through repeated use of the PF7 key.
Start of file; values read from config file	This message is given when the you press PF7 repeatedly to get to the top of the configuration file, then PF8, and then PF7.
Start of file; values read from signon defaults file	This message is given if you press PF7 repeatedly, and attempt to access data before the beginning of the file.
UNSUPPORTED FUNCTION	This message is displayed if the TSO software has not been started, or if the user typed in a command unknown to the system.
User does not have active work	This message is given if you select line commands F or C, and there is no active subtask in the USER STATUS.
USS Command Completed Successfully	This message is given when the CICS signon is successful.
Values read from config file * * * * *	This message is displayed if you press PF9, and the screen is updated with the values recorded in the configuration file.
XXXXXXXXXXXXXXXXXXXX not NETWORK NODE IN NETMAP specified	XXXXXXXXXXXXXXXXXXXX is a variable field which represents a node name. This message is given if you type in the DTF NODE NAME , and then press PF5. There is no match between the DTF NODE NAME and those specified in the NETMAP.
You must sign on to CICS before using Connect:Direct CICS Interface	This message is displayed in the STATUS ALERT MESSAGE LINE if you attempted to sign on using the CD or CDA transaction before CICS.

DBCS Support

This appendix includes the following :

- ◆ Understanding DBCS
- ◆ Default translation tables
- ◆ Customizing the translation tables

Understanding DBCS

Some languages have too many symbols for all of their respective characters to be represented using single-byte codes. For example, the English language can be defined within a single-byte range from 1-256, or x'00' through x'FF'. Hanqoul, the language of Korea, and other ideographic languages contain several thousand characters. To create these coded character sets, two bytes are needed for each character.

The Connect:Direct for VSE/ESA double-byte character set (DBCS) support provides a mechanism for translating ASCII and EBCDIC DBCS data. DBCS support will translate single-byte character set (SBCS) and DBCS data in the form that is supported on the requested platform.

DBCS Representation

The representation for DBCS characters is different between operating system platforms. Specifically, a mainframe represents data in 8-bit EBCDIC code and a PC represents data in 7-bit ASCII code. For the mainframe environment, DBCS can be used exclusively within a file or be mixed with SBCS characters. Special character indicators exist to tell the difference between SBCS and DBCS characters. The special character indicators are shift-out (SO) and shift-in (SI), or x'0E' and x'0F' respectively for IBM mainframes. Shift-out denotes shifting from SBCS to DBCS mode and shift-in denotes shifting from DBCS to SBCS mode. There is no need for SO/SI combinations if DBCS is exclusive within a file. For the PC, the SO/SI characters are not recognized. In this environment DBCS is represented by setting the high order bit of the ASCII code. See the table on page 205 for proper mapping of DBCS characters by language.

Default Translation Tables

Connect:Direct for VSE/ESA provides translation support for the Korean Standard Code Page (KS5601), DBCS-PC Korean, Chinese 5550, and Chinese Big5 to host (EBC/NHC). These tables are provided in load module and source form. See the Connect:Direct Processes Web site at www.sterlingcommerce.com/Documentation/processes/processhome.html for information on using these tables with the COPY statement SYSOPTS parameter.

The following tables are in &CD.LINKLIB:

- ◆ **EBCXJIS** translates data from host Japanese EBCDIC to ASCII Japanese.
- ◆ **JISXEBC** translates data from ASCII Japanese to host Japanese EBCDIC.
- ◆ **EBCXKSC** translates data from host EBCDIC to ASCII KS5601.
- ◆ **KSCXEBC** translates data from ASCII KS5601 to host EBCDIC.
- ◆ **EBCXKPC** translates data from host EBCDIC to DBCS-PC Korean.
- ◆ **KPCXEBC** translates data from DBCS-PC Korean to host EBCDIC.
- ◆ **NHCXBG5** translates data from Chinese new host code to Chinese Big5.
- ◆ **BG5XNHC** translates data from Chinese Big5 to Chinese new host code.
- ◆ **NHCXC55** translates data from Chinese new host code to Chinese 5550.
- ◆ **C55XNHC** translates data from Chinese 5550 to Chinese new host code.

The following source tables are in &CD.SAMPLIB:

- ◆ **EBCXJIS** translates data from host Japanese EBCDIC to ASCII Japanese.
- ◆ **JISXEBC** translates data from ASCII Japanese to host Japanese EBCDIC.
- ◆ **EBCXKSC** source for EBCDIC to ASCII KS5601.
- ◆ **KSCXEBC** source for ASCII KS5601 to host EBCDIC.
- ◆ **EBCXKPC** source for host EBCDIC to DBCS-PC Korean.
- ◆ **KPCXEBC** source for DBCS-PC Korean to host EBCDIC.
- ◆ **NHCXBG5** source for Chinese new host code to Chinese Big5.
- ◆ **BG5XNHC** source for Chinese Big5 to Chinese new host code.
- ◆ **NHCXC55** source for Chinese new host code to Chinese 5550.
- ◆ **C55XNHC** source for Chinese 5550 to Chinese new host code.

These translation tables are provided in source code format so, if needed, they can be copied or customized for your unique processing environment.

Customizing the Translation Tables

For proper translation of code pages other than the supplied translation tables, or if the supplied translation tables are not sufficient, a means of creating and updating translation tables is available.

This is done by a preprocessor that takes simple batch input in a predefined format and creates output compatible with the assembler. The output can then be assembled and link-edited to produce a loadable translation table.

Input to the batch preprocessor consists of six main parameters and the END parameter. All input begins in column one. The following table defines the batch preprocessor parameters.

Parameter	Required	Default	Format	Definition
NAME	No	XLATE	8 characters	Table Name information
TITLE	No	DBCS TRANSLATION TABLE	60 characters	Table Title information
DEFAULT	No	0000	2 byte hex representation	Default translation character
RULES	No	80-FF	2 byte hex representation	Language Rules
SBCS	No	Standard	2 byte hex representation	Single-byte character set translation table
DBCS	Yes	None	4 byte hex representation	Double-byte character set translation table
END	Yes	None		Terminates DBCS, SBCS, and RULES parameters

Required Parameters

Parameter	Description
DBCS f1f2,t1t2	<p>is used to create the double-byte character set translation table. This table will be used to translate all double-byte data during a file transfer. This parameter has no default and is required. The DBCS parameter data begins in column one and is terminated with the END statement.</p> <p>The following example shows the syntax for the RULES parameter.</p> <pre> DBCS f1f2,t1t2 END </pre> <p>f1 denotes the first byte of the FROM DBCS character. f2 is the second byte of the FROM DBCS character. t1 is the first byte of the TO DBCS character. t2 is the second byte of the TO DBCS character.</p>

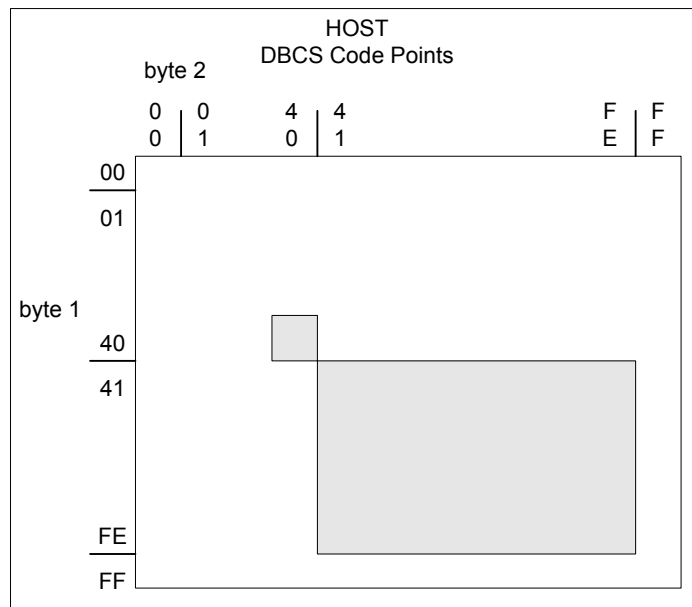
Parameter	Description
END	is mandatory to terminate each of the following parameters: DBCS RULES SBCS

Optional Parameters

Parameter	Description
NAME=[tablename <u>XLATE</u>]	<p>is an eight-character parameter for displaying table information in batch format. NAME is optional and is for informational use only. If used, NAME must be the first parameter defined. If used with TITLE, NAME and TITLE must be the first two parameters defined. The default for NAME is XLATE.</p> <p>The following example shows the syntax for the NAME parameter.</p> <pre>NAME=[tablename XLATE]</pre>
TITLE=[title name <u>DBCS TRANSLATION TABLE</u>]	<p>is a 60-character parameter for displaying table information in batch format. TITLE is optional and is for informational use only. If used, TITLE must be the first parameter defined. If used with NAME, NAME and TITLE must be the first two parameters defined. The default for TITLE is DBCS TRANSLATION TABLE.</p> <p>The following example shows the syntax for the TITLE parameter.</p> <pre>TITLE=[title name DBCS TRANSLATION TABLE]</pre>
DEFAULT=nnnn	<p>contains the hexadecimal representation you define as the replacement for invalid DBCS code points. This default character will appear wherever a non-translatable character appears in the data being received. The default is 0000.</p> <p>The following example shows the syntax for the DEFAULT parameter.</p> <pre>DEFAULT=nnnn</pre> <p>nnnn denotes the hexadecimal character defined to replace an invalid DBCS code point.</p>
RULES	<p>is used to define what constitutes a double-byte character for the defined language. RULES is only used when receiving a file from a platform other than VSE. This is because the host cannot determine valid DBCS characters without language rules. The default is any character within the range of x'80' through x'FF', meaning Connect:Direct will interpret any character within this range as the first byte of a DBCS pair. Both characters in the pair will be translated to host DBCS. If specified, use the END statement to terminate the RULES parameter.</p> <p>The following example shows the syntax for the RULES parameter.</p> <pre>RULES language option END</pre>

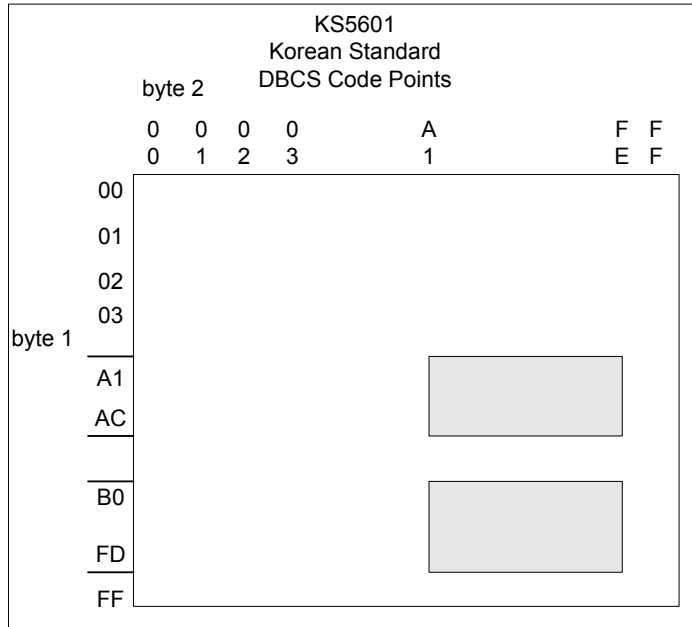
Parameter	Description														
RULES (<i>continued</i>)	Valid language options for the RULES parameter are:														
	<table border="1"> <thead> <tr> <th>Language Option</th> <th>Range</th> </tr> </thead> <tbody> <tr> <td>KS5601 (Korean Standard)</td> <td>x'A1'-x'AC' x'B0'-x'FD'</td> </tr> <tr> <td>KOREAN (Old Style)</td> <td>x'81'-x'BF'</td> </tr> <tr> <td>JAPANESE</td> <td>x'81'-x'9F' x'E0'-x'FC'</td> </tr> <tr> <td>CHINESE (Traditional/Simplified and 5550)</td> <td>x'81'-x'FC'</td> </tr> <tr> <td>BIG5 (Chinese)</td> <td>x'A4'-x'C6' x'C9'-x'F9'</td> </tr> <tr> <td></td> <td>x'01'-x'FF' user selectable</td> </tr> </tbody> </table>	Language Option	Range	KS5601 (Korean Standard)	x'A1'-x'AC' x'B0'-x'FD'	KOREAN (Old Style)	x'81'-x'BF'	JAPANESE	x'81'-x'9F' x'E0'-x'FC'	CHINESE (Traditional/Simplified and 5550)	x'81'-x'FC'	BIG5 (Chinese)	x'A4'-x'C6' x'C9'-x'F9'		x'01'-x'FF' user selectable
Language Option	Range														
KS5601 (Korean Standard)	x'A1'-x'AC' x'B0'-x'FD'														
KOREAN (Old Style)	x'81'-x'BF'														
JAPANESE	x'81'-x'9F' x'E0'-x'FC'														
CHINESE (Traditional/Simplified and 5550)	x'81'-x'FC'														
BIG5 (Chinese)	x'A4'-x'C6' x'C9'-x'F9'														
	x'01'-x'FF' user selectable														

The following figure is a graphic representation of the Connect:Direct for VSE/ESA hexadecimal DBCS code points. These code points coincide with the range values in the previous table.

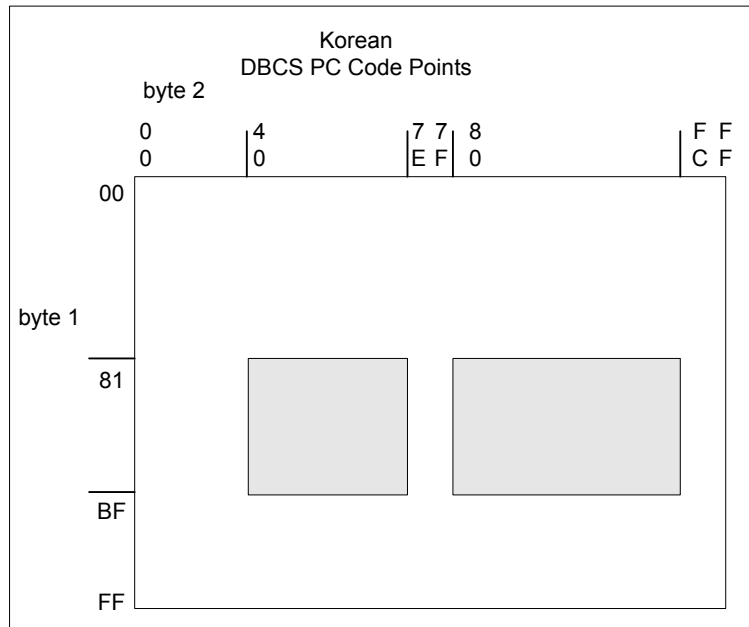


Parameter	Description
-----------	-------------

The following figure is a graphic representation of the Korean Standard (KS5601) hexadecimal DBCS code points. These code points coincide with the range values in the table on page 205.

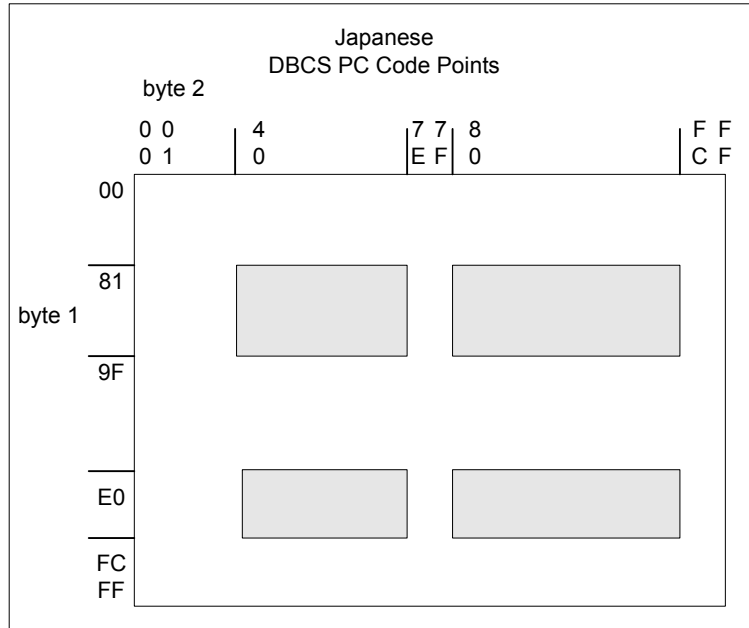


The following figure is a graphic representation of the Korean hexadecimal DBCS code points. These code points coincide with the range values in the table on page 205.

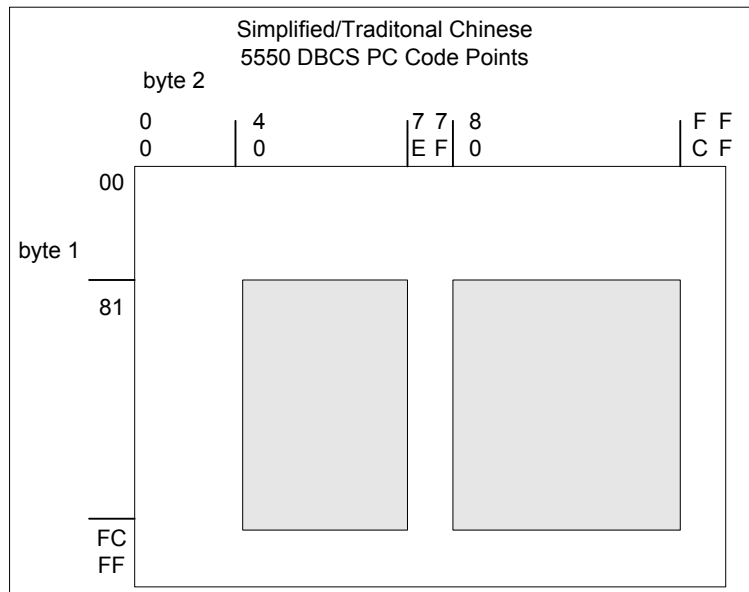


Parameter	Description
-----------	-------------

The following figure is a graphic representation of the Japanese hexadecimal DBCS code points. These code points coincide with the range values in the table on page 205.

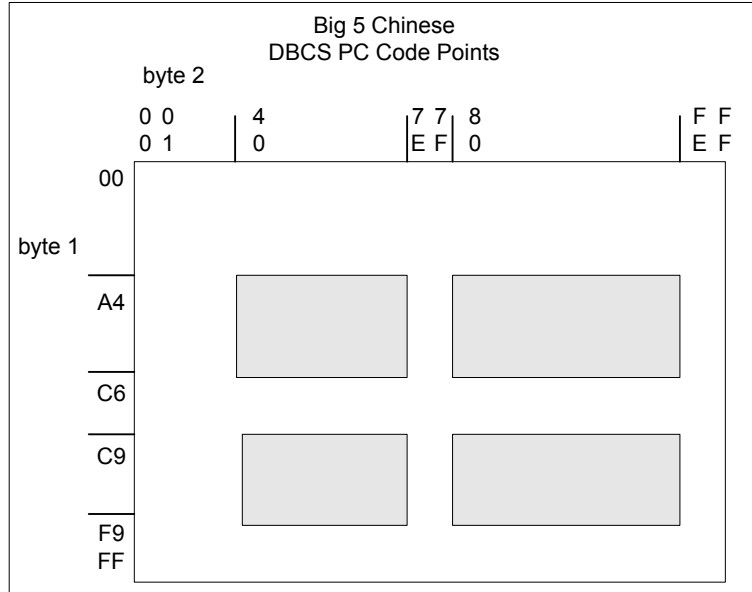


The following figure is a graphic representation of the Traditional Chinese hexadecimal DBCS code points. These code points coincide with the range values in the table on page 205.

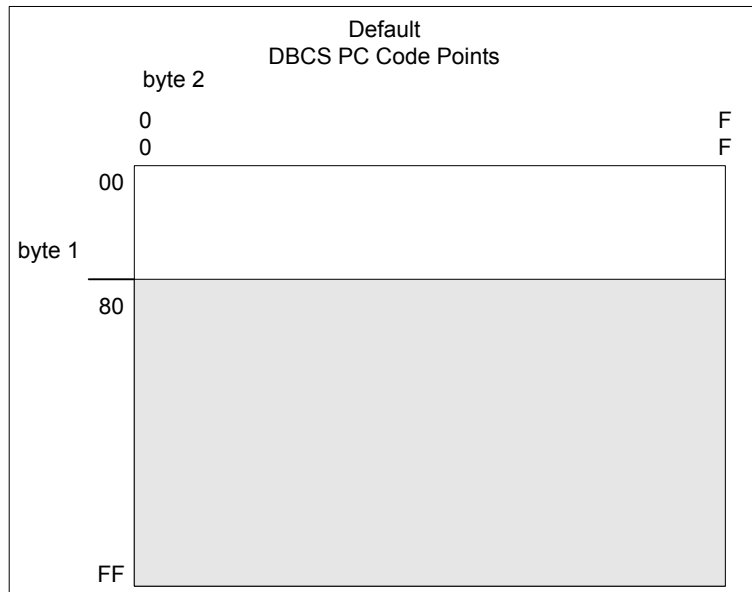


Parameter	Description
-----------	-------------

The following figure is a graphic representation of the Chinese (BIG5) hexadecimal DBCS code points. These code points coincide with the range values in the table on page 205.



The following figure is a graphic representation of the default hexadecimal DBCS code points.



Parameter	Description
SBCS ff,tt	<p>creates the single-byte character set translation table. This table is used to translate all single-byte data during a file transfer. The default translation table provided when the parameter is not specified, translates all EBCDIC characters in the range of x'00' through x'FF' to its ASCII equivalent, within the range of x'00' through x'7F'. When receiving the file from a PC, the data is translated from ASCII to EBCDIC. Terminate the SBCS parameter with the END statement.</p> <p>If SBCS is defined, all data must begin in column one and only one hexadecimal character pair is allowed per line.</p> <p>The following example shows the syntax for the SBCS parameter.</p> <pre>SBCS ff,tt END</pre> <p>ff denotes the FROM translation. tt denotes the TO translation.</p>

Comments

Comments allow you to include additional information in a batch preprocessor. Comments are available as a convenience to the user and do not affect Connect:Direct. The format for a comment is an asterisk (*) in column 1, followed by the comment. The following figure is a sample comment with an asterisk in column 1.

```
* DEFAULT=FFFF instead of 0000.
```

Using the RULES Parameter

The following example will translate all characters as DBCS that adhere to KS5601 standard, or all characters that start with an x'A1' through x'AC' or x'B0' through x'FD'. Treat these characters as double-byte characters.

```
RULES
KS5601
END
```

The following example will translate all characters as DBCS that adhere to the customized table. Treat all characters that start with x'90' through x'94' or x'B0' through x'B4' as double-byte characters.

```
RULES
90
91
92
93
94
B0
B1
B2
B3
B4
END
```

Using the SBCS Parameter

The following example will translate x'C1' to x'41', x'C2' to x'42', x'C3' to x'43', and so on.

```
SBCS
C1,41
C2,42
C3,43
C4,44
C5,45
C6,46
END
```

Using the DBCS Parameter

A DBCS table can be extremely large and complex; therefore, the following examples are sample tables. The examples should be used as a reference only and will not successfully translate all characters.

The following example translates x'89A1' to x'B0ED', x'89A2' to x'B0EE', x'89A5' to x'B0EF', and so on to x'D37B' to x'C8F0'.

```
DBCS
89A1,B0ED
89A2,B0EE
89A5,B0EF
89A8,B0F0
89A9,B0F1
89AA,B0F2
89AB,B0F3
D375,C8EE
D377,C8EF
D37B,C8F0
END
```

Sample Preprocessor Input Data Stream

The following sample is the syntax for a preprocessor input data stream. The SBCS and DBCS tables are incomplete and would require many pages to produce a valid table.

```
NAME=MYTABLE
TITLE=SAMPLE TRANSLATION TABLE
RULES
80
81
82
83
84
END
SBCS
C1,41
C2,42
C3,43
C4,44
C5,45
C6,46
END
DBCS
89A1,B0ED
89A2,B0EE
89A5,B0EF
89A8,B0F0
89A9,B0F1
89AA,B0F2
89AB,B0F3
D375,C8EE
D77,C8EF
D37B,C8F0
END
```

Sample JCL to EXECUTE the Preprocessor

The following sample JCL executes the preprocessor against the input source. The output produced by the preprocessor is in assembler CSECT form and is input to the assembler. The assembled object will then be link-edited to produce a load module.

The following JCL is contained in the member DBCSBLD \$CD.INSTALL.JCL:

```

* $$ JOB JSASME,D,3,0
* $$ LST DEST=(,JSTAN1)
* $$ PUN DEST=(,JSTAN1)
// JOB JSASMLNK
// EXEC PROC=CDLIBS,REL=2100
// DLBL IJSYSUC,'VSESP.USER.CATALOG',,VSAM
// ASSGN SYS020,DISK,VOL=USER01,SHR
// DLBL IJSYSPH,'VSE.SYSPUNCH,'
// EXTENT SYSPCH,USER01,1,0,xxxx,yyyy
// ASSGN SYSPCH,SYS020,PERM
// EXEC DMDBCSPR
CONTROL STATEMENTS USED TO CREATE 'DBCSTBL'
/*
CLOSE SYSPCH,PUNCH
// OPTION CATAL
PHASE DBCSTBL,*
// DLBL IJSYSIN,'VSE.SYSPUNCH',,SD
// EXTENT SYSIPT
// ASSGN SYSIPT,SYS020,PERM
// EXEC ASSEMBLY
/*
CLOSE SYSIPT,FEC
// EXEC LNKEDT,PARM='MSHP,AMODE=ANY'
/*
/&
* $$ EOJ

```

In this example, **xxxx** is the STARTING TRACK for the EXTENT and **yyyy** is the number of TRACKS for the EXTENT.

See the Connect:Direct Processes Web site at www.sterlingcommerce.com/Documentation/processes/processhome.html for information on utilizing the table created from the previous JCL sample.

A

ADJACENT.NODE

Adjacent node entries in the network map 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), Connect:Direct Batch Interface, the console operator interface, or user-written program and places the commands 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 APPLID is 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 interactive or batch API APPLID.

API System ID(s)

This ID is the system identifier (SYSIDs or SMF IDs) of the CPUs (up to 16) that will be sharing a copy of the Connect:Direct for VSE/ESA DTF utilizing the Shared DASD Facility (SDF) option of Connect:Direct.

Authorization Facility

Connect:Direct authorization facility grants access to Connect:Direct commands.

Authorization File

Connect:Direct for VSE/ESA authorization file controls access to Connect:Direct for VSE/ESA and identifies commands that can be executed by the userid. This file can also be used in conjunction with security exit interfaces with the secure point-of-entry feature.

B

Batch Interface

This Connect:Direct interface allows users to request Connect:Direct services from a batch jobstream through control statements passed to a Connect:Direct-supplied program, DMBATCH.

C

Checkpoint/Restart

For the COPY statement, 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 parameters, CKPT.MODE and CKPT, specifies the checkpoint interval. If a copy procedure is interrupted, Connect:Direct will restart that copy at the last checkpoint.

For the RUN TASK Statement, checkpoint/restart determines whether the RUN TASK program is to be executed again at Process restart if Connect:Direct is unable to determine whether the program has executed. RESTART in the RUN TASK statement or the RUNTASK.RESTART initialization parameter specifies whether to restart the program at the last checkpoint.

Connect:Direct Commands

These commands initiate and monitor activity within the Connect:Direct system. You can issue commands from the IUI, the operator console, a batch job or a user application program.

Connect:Direct Process

The Connect:Direct Process is a series of statements submitted through the API to initiate Connect:Direct for VSE/ESA activity, such as copying files and running jobs. The Processes can be predefined and stored in a library.

Cross Domain

Cross domain pertains to controlling resources that involve more than one domain. A domain consists of one SSCP (System Services Control Point) and all the SNA resources under its control. This term applies when you perform a multiple-session signon to another Connect:Direct DTF and when you establish a connection to another Connect:Direct DTF to copy files.

D

DTF (Data Transmission Facility)

The DTF is 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

ESF (Extended Submit Facility)

The ESF allows users to queue data transfer requests to a Connect:Direct node that is not active. This allows users to submit work to Connect:Direct, even when the Connect:Direct DTF is down.

I

IUI (Interactive User Interface)

The IUI is a screen and dialog component that allows users to define and submit Connect:Direct Processes and issue Connect:Direct commands that monitor and control administrative and operations activity. An IUI is available for a CICS environment with the CICS interface provided with the Connect:Direct for VSE/ESA product.

L

LOCAL.NODE

The local node entry defines the logical Connect:Direct name of the local Connect:Direct DTF and its associated communications name. The local node entry also contains the name of the transmission queue and the SUPERUSR ID password, if specified. The local node entry is in the network map.

M

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 the completion of a previous Process step.

N

Network Map

VSAM file identifying all valid Connect:Direct nodes and applids in the network. Each Connect:Direct for VSE/ESA node has a network map. There is one entry within that network map for each of the other Connect:Direct nodes to which the local Connect:Direct for VSE/ESA node can initiate a session. The entries also contain the rules or protocol the nodes use when communicating.

Node

Any site within a network from which information distribution is initiated.

O

Online Messages

Completion and error messages that are displayed online.

Operator Interface

Allows Connect:Direct commands to be issued from the VSE operator console. This interface also allows tailoring of Connect:Direct commands through a command list (CLIST) facility.

P

Parallel Sessions

Parallel sessions allow you to have 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)

The PNODE is the Connect:Direct node on which the Process is being submitted. The primary node may also be referred to as the controlling or source node, but should not necessarily be interpreted as the sending node since PNODE can be the receiver. In every Process, there is one PNODE and one SNODE specified. The submitter of a Process is always the PNODE. The PNODE name can be expressed in 1-16 alphanumeric and national characters (@ # \$), with imbedded periods. The first character must be alphabetic.

Process

A Process is a series of statements (which can be predefined and stored in a library) submitted through the API to initiate Connect:Direct for VSE/ESA activity, such as copying files and running jobs.

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).

S

SNA (Systems Network Architecture)

SNA is a network architecture designed to provide compatibility among a wide variety of hardware and software products so that they can be used to build complex networks. It defines protocols, standards, and message formats to which different hardware and software products must conform.

SNODE (Secondary Node)

The SNODE is a 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

This file is a VSAM relative record data set that holds Connect:Direct statistics records to document the history of a Process. This file is a wrap-around file.

Statistics Facility

The Connect:Direct statistics facility records all Connect:Direct activities

SYMBOL Statement

This Connect:Direct Process statement allows you to build symbolic substitution values.

Symbolics

Connect:Direct Processes support symbolic parameters. The parameters allow one predefined Process to be used for multiple applications. For example, the file names for a COPY could be passed to the Process by the user submitting the Process.

T

TCQ (Transmission Control Queue)

The TCQ is a VSAM relative record data set (RRDS) used to hold all Processes that have been submitted to Connect:Direct for VSE/ESA.

TCP/IP (Transmission Control Protocol/Internet Protocol)

TCP/IP 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.

Symbols

\$\$ACTION verbs 81
\$\$BLKxxxxxx 82
\$\$DELETE 82, 84
\$\$ENDSYNTAX 82
\$\$ENDVERIFY 82
\$\$ENDxxxxxx 82
\$\$INSERT 81, 83
\$\$SYNTAX 82, 85
\$\$UPDATE 82, 83
\$\$VERIFY 82, 86

A

A (Activate (start first task)) line command 183
A (ACTIVATE INTERFACE) entry field 164
ABENDS, task 166
accessible nodes 185
accounting information 167
action verbs
 example of \$\$DELETE 84
 example of \$\$INSERT 83
 example of \$\$SYNTAX 85
 example of \$\$UPDATE 83
 example of \$\$VERIFY 86
 examples 83
 updating Network Map 81
Activate (start first task) (A) line command 183
ACTIVATE ADMIN REQUEST 184
ACTIVATE INTERFACE (A) entry field 164
ACTIVE CICS STATUS 184
ACTIVE TASKS field 164

ADD TYPE parameter
 with INSERT USER 48
 with UPDATE USER 53
ADD USER parameter
 with INSERT USER 48
 with UPDATE USER 53
adding a DTF node record 177
adding a network node record 179
adding a user sign-on record 193
adjacent node
 communications name 67
 data direction restriction 69
 ENVIRONMENT parameter 70
 IP address or Alias 68
 keyword parameter 69
 LOGMODE parameter 70
 NDMPACE parameter 71
 NETID parameter 70
 PNODE.LUS parameter 70
 positional parameters 66
 remote library name 68
 security node type 68
 session type 68
 SNODE.LUS parameter 71
adjacent node example
 MS-DOS LU0 74
 MS-DOS LU6.2 74
 MVS LU6.2 73
 MVS SNA LU0 72
 MVS Trusted Node 73
 NetWare LU6.2
 dependent LUs 75
 independent and dependent LUs 75
 independent LU name designated 74
 NetWare TCP/IP 75
 OpenVMS 74
 OS/2 LU0 76
 OS/2 LU6.2
 dependent LU 76
 independent LU 76

Index

- OS/400 LU6.2
 - dependent LU 77
 - independent LU 77
 - OS/400 SNUF 76
 - TCP/IP 73
 - UNIX TCP/IP 77
 - VM SNA LU0 78
 - VSE 78
 - Windows NT TCP/IP 79
 - ADJACENT.NODE keyword 66
 - ADMIN REQUEST field 183
 - alias 68
 - allocation exit 106
 - addresses and values, calculating 107
 - Copy Control Block
 - Definitions 108
 - Modifications 108
 - DDESCR Control Block Format 109
 - modifiable fields 114
 - execution 106
 - invoking 106
 - parameters 106
 - sample 106
 - use with I/O exit 116
 - ALLOCATION.EXIT 106
 - ALTER TYPE parameter
 - with INSERT USER 48
 - with UPDATE USER 53
 - ALTER USER parameter
 - with INSERT USER 48
 - with UPDATE USER 53
 - API (application program interface) 186
 - Connect:Direct DTF 162
 - endiui 153
 - high-level interface 142, 143
 - sample program 196
 - session trace 28
 - application program, writing 142
 - applid for DTF-DTF communications
 - adjacent Node 67
 - local node 65
 - ARCHDSN parameter 129
 - architecture, Connect:Direct for CICS 153
 - archiving
 - DMSTARRT sample 131, 132
 - sample setup 133
 - SELECT STATISTICS command 129
 - statistics 128
 - statistics files 26
 - utilities 130
 - authorization
 - bit mask 166, 175
 - facility 88, 165
 - file 175
 - user 45
 - authorization file
 - example 93
 - purpose 45, 93
 - AUTO.SIGNON entry field 173
 - autostart, Connect:Direct for CICS 161
- ## B
- batch interface
 - displaying tasks 16
 - removing tasks from execution 15
 - BG5XNHC
 - source table 204
 - translation table 204
 - bit mask 166, 175
 - BITS.OFF parameter 28
 - BITS.ON parameter 28
 - BLK 82
- ## C
- C (CONFIGURATION) option 160
 - C (CONTROL RECORD) option 172
 - C (FREE USER AND CANCEL USER'S SUBTASK)
 - line command 187
 - C55XNHC
 - source table 204
 - translation table 204
 - CA-ACF2 87
 - CASE parameter 34, 39, 43, 49, 54, 57, 59
 - CASE sensitivity 192

- CDA
 - transaction 159, 166
 - TRANSACTION ENDED message 159
- CF (COPY FILE) menu option 173, 175
- Change parameter
 - with INSERT USER 49
 - with UPDATE USER 54
- CHANGE SIGNON (SN) menu option 173, 176
- Chinese 5550 204
- Chinese Big5 204
- CICS
 - CEBR transaction 197
 - CICS.TRANSACTION.CODE (MONITOR) entry field 173
 - CICS.TRANSACTION.CODE (PRINT) entry field 173
 - CICS.TRANSACTION.CODE (STARTUP) entry field 173
 - ID field 185
 - monitor transaction 161
 - PRINTER entry field 191, 193
 - STATUS system field 183
 - TERM field 185
 - TERMIN field 160
 - USERID field 160, 188
- CICS ID field 185, 187
- CICS PRINTER entry field 191, 193
- CICS STATUS field 183
- CICS TERM field 185, 187
- CICS USERID entry field 192
- CICSID keyword 165
- COMMAND entry field 195
- command format
 - DELETE TYPE command 42
 - INSERT TYPE 33
 - SELECT TYPE command 43
 - UPDATE TYPE command 38
- COMMAND MSG field 195
- COMMAND MSG ID field 195
- COMMAND RC field 195
- commands
 - administrative 163, 166
 - DELETE TYPE 31, 42
 - DELETE USER 56
 - EXEC CICS LINK 197
 - FLUSH TASK 15
 - INSERT TYPE 31, 33
 - INSERT USER 46
 - MODIFY 27
 - SELECT TASK 14, 16
 - SELECT TYPE 31, 43
 - SELECT USER 58
 - STOP CD 14, 29
 - UPDATE NETMAP 80
 - UPDATE TYPE 31, 37
 - UPDATE USER 51
 - line 183, 187
 - SUBMIT 100
- COMMAREA 195, 197
- comments, DBCS support 211
- communications name
 - adjacent node 67
 - local node 65
- components, Connect:Direct for CICS 153
- CONFIGURATION (C) option 160
- CONFIGURATION datasets 169
- configuration screen, Connect:Direct for CICS Administration 156
- Connect:Direct
 - API 186
 - Authorization Facility 165
 - description 11
 - DTF API 162
 - DTF log on 185
 - EQ.CICSID entry field 172
 - maintenance 123
 - message file 124
 - NODE field 160
 - PASSWORD entry field 192
 - PASSWORD MESSAGE field 193
 - security exit 175
 - SIGNON parameters 167
 - user profile dataset 193
 - USERID entry field 192
- Connect:Direct for CICS ADMINISTRATION CONFIGURATION screen 156

- ADMINISTRATION CONTROL RECORD
 - UPDATE screen 156
 - ADMINISTRATION DTF NODE RECORDS
 - screen 156
 - ADMINISTRATION INTERFACE screen 156
 - ADMINISTRATION NETWORK NODE
 - RECORDS screen 172
 - ADMINISTRATION NODE STATUS screen 156
 - ADMINISTRATION PRIMARY
 - menu 156, 160, 177
 - ADMINISTRATION SIGNON DEFAULTS
 - screen 156
 - ADMINISTRATION USER STATUS screen 156
 - ADMINISTRATION WORK QUEUE screen 156
 - architecture 153
 - autostart 161
 - components 153
 - control record 172
 - DTF node records 176
 - IUI 165
 - CONTROL RECORD (C) option 172
 - CONTROL.RECORD updating 176
 - controller subtask (CST) 161
 - controlling accessible nodes 185
 - COPY FILE (CF) menu option 173, 175
 - copy file between nodes screen 179
 - COPY parameter
 - with INSERT USER 49
 - with UPDATE USER 54
 - COPY routine trace 28
 - CSSF LOGOFF 159
 - CSSN transaction 159
 - CST (controller subtask) 161
 - CST.RETRY.INTERVAL entry field 172
 - CURR CMD field 185
 - CURR TASKS field 183
 - CURR WORK field 183
 - customer memo 123
- D**
- D (DTF NODE RECORDS) option 172
 - data direction restriction 69
 - Data Transmission Facility (DTF) description 153
 - date field 157
 - dates
 - how Connect:Direct processes 22
 - specifying 22
 - DBCS
 - comments 211
 - preprocessor parameter 205, 212
 - support 203
 - using SO/SI 203
 - DBCSBLD member 214
 - DBCS-PC Korean 204
 - DCB parameter
 - with Insert Type command 35, 39
 - DD names 28
 - debug settings 28
 - DEFAULT NODE entry field 191
 - DEFAULT preprocessor parameter 206
 - DELETE
 - adjacent node 82
 - command example 84
 - DELETE TYPE command
 - batch interface use 43
 - command format 42
 - information 31
 - optional parameters 42
 - purpose 31, 42
 - required parameters 42
 - DELETE USER command
 - command format 56
 - optional parameters 57
 - purpose 56
 - required parameters 57
 - using batch interface 57
 - deleting a DTF node record 177
 - deleting a network node record 180
 - deleting a user sign-on record 194
 - DELPR parameter
 - with INSERT USER 49
 - with UPDATE USER 54

DIS parameter 81
 DISP parameter 35, 40
 distribution files 98
 DMCHLAPI 141, 142
 UICB extract feature for debugging, using 144
 interprets parameters, how 150
 parameter requirements 143
 return codes 151
 sequence 142
 DMCXSIGN 92
 DMGALOEX sample allocation exit 106
 DMGCBSUB trace 28
 DMGSECUR macro 165
 DMSTARBT utility 131
 DMSTARRT utility 130
 DMSTBKEY utility 139
 description 132
 sample jobstream 133
 DTF (data transmission facility) 11, 153
 DTF NODE NAME field 178, 187
 DTF node record
 adding 177
 deleting 177
 editing 177
 reading 177
 DTF NODE RECORDS (D) option 172
 DTF node records screen 156
 DTF security exit 166
 DUMMY ID FOR DTF SIGNON entry field 178

E

EBCXJIS
 source table 204
 translation table 204
 EBCXKPC
 source table 204
 translation table 204
 EBCXKSC
 source table 204
 translation table 204

editing a DTF node record 177
 editing a network node record 180
 editing a user sign-on record 194
 EIB (Exec interface block) 196
 END preprocessor parameter 206
 ENTRIES IN WORK QUEUE entry field 177, 178
 ENVIRONMENT parameter 70
 ESF (extended submit feature) mode
 description 153, 163
 ESF MODE ALLOWED entry field 191
 ESF session 186
 ESF SIGNON ALLOWED entry field 178
 ESF.RETRY.INTERVAL entry field 173, 174
 Exec interface block (EIB) 196
 exits
 allocation 106
 DTF security 166
 I/O 116
 sample 106
 Security 95
 security 175
 statistics 95
 Submit 95, 100
 Stage 1 101
 Stage 2 101
 User 95
 Extended Submit Feature (ESF)
 mode description 153, 163
 with DMBATCH 50, 55

F

F (FREE USER) line command 187
 file
 authorization 175
 NETMAP 165, 178
 TCQ (transmission control queue) 168
 Trace 168
 file pairs 127
 change restrictions 139
 changing configuration 139
 list changes 139
 number parameter 26

Index

- FLUSH parameter
 - with INSERT USER 49
 - with UPDATE USER 54
 - FLUSH TASK command
 - syntax 15
 - through batch interface 15
 - FORCE parameter
 - with FLUSH TASK command 15
 - with STOP CD command 29
 - FREE USER (F) line command 187
 - FREE USER AND CANCEL USER'S SUBTASK (C)
line command 187
 - FREE USER AND TERMINATE USER'S SIGNON (T)
line command 187
- ### G
- GETMAIN/FREEMAIN trace 28
- ### I
- I (INTERFACE STATUS) option 160
 - I (Shut immediate) line command 184
 - I (SHUTDOWN INTERFACE (IMMEDIATE)) entry
field 164
 - I/O buffer trace 28
 - I/O exit 116
 - access to control blocks 117
 - COPY statement, specification on 116
 - implementing 116
 - COPY statement 116
 - TYPE file 117
 - IOEXIT keyword
 - COPY Statement 116
 - TYPE File commands 117
 - normal calling sequence
 - input 121
 - output 121
 - purpose 116
 - requests 119
 - ADD 120
 - BEGIN 119
 - CLOSE 120
 - END 120
 - GET 120
 - INFO 120
 - OPEN 119
 - sample 116
 - TYPE file, specification in 117
 - ID, CICS terminal 188
 - Immediate parameter 29
 - INACT CICS STATUS 184
 - INACTIVE.INTERVAL entry field 173, 175
 - initialization parameters
 - display settings 16
 - SECURITY 88
 - STAT.FILE.PAIRS 127
 - SUBMIT.EXIT 100
 - INQUIRE INITPARMS command 16
 - Batch Interface usage 17
 - parameters 17
 - INQUIRE STATDIR command 21
 - Batch Interface usage 23
 - parameters 22
 - with statistics 130
 - INQUIRE STATISTICS command 20
 - Batch Interface usage 20
 - INQUIRE STATISTICS Command 137
 - parameters 20
 - INSERT
 - adjacent node 81
 - command example 83
 - INSERT TYPE command
 - batch interface usage 37
 - command format 33
 - information 31, 33
 - optional parameters 34
 - purpose 31, 33
 - required parameter 34
 - INSERT USER command
 - authorization record parameters 48
 - ADD TYPE 48
 - ADD USER 48
 - ALTER TYPE 48
 - ALTER USER 48
 - READ TYPE 48
 - READ USER 48
 - REMOVE TYPE 48
 - REMOVE USER 49
 - example 51
 - format 46

- functional authorization parameters 49
- optional parameters 48
- purpose 46
- required parameters 48
- using batch interface 50

interactive user interface (IUI)

- Connect:Direct for CICS 165
- description 153
- signing on to a single DTF 165

interface

- request management 162
- startup 162
- subtask management 162
- tasks 162
- VTAM session 163

INTERFACE STATUS (I) option 160

INTERFACE STATUS field 164

interfacing with Connect:Direct, DMCHLAPI 141

internal components 11

- Application Program Interface 11
- Data Transmission Facility
 - data transmission facility 11
- interfaces 11

IOEXIT keyword

- COPY Statement, on 116
- TYPE File commands, on 117

IP address 68

IUI (interactive user interface)

- iu (interactive user interface)
 - Connect:Direct for CICS 165
 - description 153
 - signing on to a single DTF 165

J

JISXEBC 204

K

keyword

- ADJACENT.NODE 66
- CICSID 165
- keyword parameters
 - adjacent node 69
 - local node 65
- LOCAL.NODE 64, 124

Korean Standard Code Page 204

KPCXEBC

- source table 204
- translation table 204

KS5601 Korean Standard Code Page 204

KSCXEBC

- source table 204
- translation table 204

L

LAST REQUEST field 188

LAST SIGNON field 188

LAST TASK# field 185, 187

line commands 183, 187

local node

- communications name 65
- keyword parameter 65
- LOCAL.NODE keyword 64
- node name 64
- positional parameters 64
- superuser password 65
- TCQ parameter 65
- updating Network Map 80

LOCAL.NODE keyword 124

log on to Connect:Direct DTF 185

logging information 167

LOGMODE parameter 70

logon exit trace 28

logon processor trace 28

LU0 adjacent node example

- MVS SNA 72
- OpenVMS 74
- Tandem 77

LU6.2 adjacent node example

- MS-DOS 74
- MVS 73
- NetWare
 - dependent LUs 75
 - independent and dependent LUs 75
 - independent LU name designated 74

OS/2
 dependent LU 76
 independent LU 76
 OS/400 76
 dependent LU 77
 independent LU 77
 UNIX 78
 VSE 79

M

M (START MONITOR) entry field 164
 macro, DMGSECUR 165
 maintenance 123
 management, interface request 162
 management, interface subtask 162
 map, network 162
 mask bit 175
 MAX TASKS field 183
 MAX.SIGNON entry field 173, 175
 MAX.TASKS entry field 173, 175
 MAXIMUM WORKER SUBTASKS entry field 177, 178
 MAXSA parameter
 with INSERT USER 49
 with UPDATE USER 54
 MD (MESSAGE DISPLAY) menu option 173, 176
 menu options
 CHANGE SIGNON (SN) 173, 176
 COPY FILE (CF) 173, 175
 MESSAGE DISPLAY (MD) 173, 176
 SELECT PROCESS (SP) 173, 175
 SELECT STATISTICS (SS) 173, 175
 SIGNON DEFAULTS (SD) 173, 176
 SUBMIT PROCESS (SB) 173, 175
 MESSAGE dataset 168
 MESSAGE DISPLAY (MD) menu option 173, 176
 message field 157
 message file 124
 message, CDA TRANSACTION ENDED 159
 MOD field 163

MODALS parameter 49, 54
 MODDIR.TRACE parameter 27
 mode, Extended Submit Feature (ESF) description 153
 MODETAB 70
 MODIFY command
 Batch Interface usage 28
 example, batch 28
 format 27
 parameters 28
 purpose 27
 MODIFY parameter
 with INSERT USER 49
 with UPDATE USER 54
 MONITOR TASK NUMBER field 164
 MONITOR.INTERVAL entry field 173, 175
 monitoring statistics 136
 monitoring user status 188
 MS-DOS LU0 adjacent node example 74
 MS-DOS LU6.2 adjacent node example 74
 MS-DOS node type 180, 181
 MSGID field 187
 multiple Connect:Direct for CICS IUI facilities 165
 MVS LU6.2 adjacent node example 73
 MVS node type 179, 181
 MVS SNA LU0 adjacent node example 72
 MVS trusted node adjacent node example 73

N

N (NETWORK NODE RECORDS) option 172
 N (NODE STATUS) option 160
 NAME parameter
 with INSERT USER 48
 with UPDATE USER 53, 54
 NAME preprocessor parameter 206
 NDMPACE
 adjacent node 71
 adjacent node keyword 71
 NETID 70
 NETINPUT parameter 81

- NETLOG parameter 81
 - NETMAP DDNAME entry field 178
 - NETMAP file 165, 168, 178
 - NetWare LU6.2 adjacent node example
 - dependent LUs 75
 - independent and dependent LUs 75
 - independent LU name designated 74
 - NETWARE node type 180, 181
 - NetWare TCP/IP adjacent node example 75
 - NETWORK DLBL name 178
 - network map 162
 - adjacent node
 - communications name 67
 - data direction restriction 69
 - ENVIRONMENT parameter 70
 - IP address or Alias 68
 - keyword parameters 69
 - LOGMODE parameter 70
 - NDMPACE parameter 71
 - NETID parameter 70
 - PARSESS parameter 69
 - PNODE.LUS parameter 70
 - positional parameters 66
 - remote library name 68
 - security node type 68
 - session type 68
 - SNODE.LUS parameter 71
 - batch update 82
 - local node
 - communications name 65
 - keyword parameters 65
 - node name 64
 - positional parameters 64
 - superuser password 65
 - TCQ parameter 65
 - local node updating 80
 - Size 61
 - update dynamically 80
 - update while Connect:Direct is down 80
 - updating 79
 - NETWORK NODE DESCRIPTION entry field 181
 - NETWORK NODE NAME entry field 181
 - network node record
 - adding 179
 - deleting 180
 - editing 180
 - reading 180
 - NETWORK NODE RECORDS (N) option 172
 - network node records screen, Connect:Direct for CICS Administration 156, 172
 - NHCXBG5
 - source table 204
 - translation table 204
 - NHCXC55
 - source table 204
 - translation table 204
 - NODE field 185
 - node name 64
 - NODE STATUS (N) option 160
 - node status screen, Connect:Direct for CICS Administration 156
 - NODE TYPE entry field 181
 - node types
 - MS/DOS 181
 - MS-DOS 180
 - MVS 179, 181
 - NETWARE 180
 - OS/2 180, 181
 - OS/400 180, 181
 - Tandem 179, 181
 - UNIX 180
 - VM 179, 181
 - VMS 179, 181
 - VSE 179, 181
 - Windows NT 180
 - NODE, Connect:Direct 160
 - nodes, accessible 185
- O**
- OpenVMS adjacent node example 74
 - OPTION fields
 - Configuration Screen 171
 - Interface Screen 164
 - Primary Menu 160
 - options
 - CONFIGURATION (C) 160
 - CONTROL RECORD (C) 172
 - DTF NODE RECORDS (D) 172

Index

INTERFACE STATUS (I) 160
NETWORK NODE RECORDS (N) 171
NODE STATUS (N) 160
SIGNON DEFAULTS (S) 160
USER STATUS (U) 160

OS/2 LU0 adjacent node example 76
OS/2 LU6.2 adjacent node example
 dependent LU 76
 independent LU 76
OS/2 node type 180, 181
OS/400 LU6.2 adjacent node example
 dependent LU 77
 independent LU 77
OS/400 node type 180, 181
OS/400 SNUF adjacent node example 76
OUTPUT RECORD LIMIT entry field 177, 178

P

PAGE XX OF YY field 185
parallel sessions 69
PARSESS parameter 69
PASSword parameter
 with INSERT USER 49
 with UPDATE USER 54
PENDING REQUEST field 164
PF key assignments 157
PHone parameter
 with INSERT USER 49
 with UPDATE USER 54
PNODE 185
PNODE ACCT DATA entry field 191, 193
PNODE.LUS 70
PNODE=SNODE processing
 adjacent node entry required 72
 concept 72
 PARSESS parameter required 69
port number 68
positional parameters
 adjacent node 66
 local node 64

preprocessor
 batch input 205
 input data stream 213
 JCL to execute 213
 parameters 205
preprocessor parameter
 DBCS 205, 212
 DEFAULT 206
 END 206
 NAME 206
 RULES 206, 211
 SBCS 211, 212
 TITLE 206
PRIM SESS TYPE 184
primary menu, Connect:Direct for CICS
 Administration 156, 160, 177
Process execution
 control points 88
 detail of security flow 91
 security control points 91
PROCESS NUMBER field 195
process PNODE 185
PRT parameter 81
pseudo DLBL 28
PUF field 163

Q

QUIESCE command 163
Quiesce parameter 30

R

RACF 87
RC (return code) 195
READ TYPE parameter
 with INSERT USER 48
 with UPDATE USER 53
READ USER parameter
 with Insert User 48
 with Update User 53
reading a DTF node record 177
reading a network node record 180

reading a user sign-on record 194

REL field 163

remote library name 68

REMOVE TYPE parameter
with Insert User 48
with Update User 53

REMOVE USER parameter
with Insert User 49
with Update User 53

REQUEST DATE/TIME field 185

RESETSA parameter 54

Resource Access Control Facility (RACF) 87

return code (RC) 195

RPL trace
long 28
short 28

RULES
preprocessor parameter 206
examples 211

RUNJOB parameter
with Insert User 50
with Update User 55

RUNTASK parameter
with Insert User 50
with Update User 55

S

S (Shut normal) line command 184

S (SHUTDOWN INTERFACE) (NORMAL) entry
field 164

S (SIGNON DEFAULTS) option 160

S2 statistics records 137

sample program, ARCHSTAT 129

SB (SUBMIT PROCESS) menu option 173, 175

SBCS
example 212
preprocessor parameter 211

SCCSTAT utility 138

SCIP exit trace 28

SD (SIGNON DEFAULTS) menu option 173, 176

security
authorization bit mask 90
exits
four types 88
implementation of 92
Stage 1 signon 92, 93
initialization parameter 88
planning for 87
Process execution
control points 88
detail of flow 91
requirements, Authorization Facility 45, 93
SIGNON command
control points 88
detail of flow 89
through Authorization Facility 87
through CA-ACF2, RACF, and TOP SECRET 87
through Run Job exit 87
through Run Task exit 87

security exits 95, 166, 175

security node type 68

SECURITY parameter
with Insert User 50
with Update User 55

security system bypass 65

SELECT PROCESS (SP) menu option 173, 175

SELECT STATISTICS (SS) menu option 173, 175

SELECT TASK command
syntax 16
through batch interface 16

SELECT TYPE command
Batch Interface usage 44
command format 43
information 31
optional parameters 44
purpose 31, 43
required parameters 44

SELECT USER command
command format 58
optional parameters 59
purpose 58
required parameters 58
using batch interface 59

Index

- SELNET parameter
 - with Insert User 50
 - with Update User 55
- SELPR parameter
 - with Insert User 50
 - with Update User 55
- SELSTAT parameter
 - with Insert User 50
 - with Update User 55
- separate trace per task trace 28
- SESS TYPE field 183, 187
- session class 105
- SESSION DATE/TIME OR SESSION MSGID field 183
- session manager 153
- session manager trace 28
- session type 68
- session, ESF 186
- SESSION.RETRY.INTERVAL entry field 172, 173, 174
- Shut immediate (I) line command 184
- Shut normal (S) line command 184
- SHUTDOWN command 163, 166
- SHUTDOWN IMMEDIATE command 166
- SHUTDOWN INTERFACE (IMMEDIATE) (I) entry field 164
- SHUTDOWN INTERFACE (NORMAL) (N) entry field 164
- SHUTIMM ADMIN REQUEST 184
- SHUTNORM ADMIN REQUEST 184
- signing off 159
- signing on 159
- SIGNON command
 - detail of security flow 89
 - execution security control points diagram 89
 - security control points 88
- SIGNON DEFAULTS (S) option 160
- SIGNON DEFAULTS (SD) menu option 173, 176
- signon defaults screen, Connect:Direct for CICS Administration 156
- SIGNON REENTRY entry field 172
- SKIP.SIGNON.PANEL entry field 172
- SLOW RESPONSE NOTIFICATION entry field 177, 178
- SN (CHANGE SIGNON) menu option 173, 176
- SNODE ACCT DATA entry field 191, 193
- SNODE.LUS 71
- SP (SELECT PROCESS) menu option 173, 175
- SPACE parameter 36, 41
- specifying dates into Year 2000 22
- SS (SELECT STATISTICS) menu option 173, 175
- SS statistics records 138
- Stage 1 signon security exit 92, 93
- Stage 1 Submit exit 101
- Stage 2 Submit exit 101
- start first task, Activate (A) line command 183
- START MONITOR (M) entry field 164
- STARTT parameter 22
- startup, interface 162
- STAT.ARCH.CONFIRM initialization parameter 129, 130
- STAT.DSN.BASE initialization parameter 127
- STAT.FILE.PAIRS initialization parameter 139
- STAT.INIT initialization parameter 140
- STAT.SWITCH.SUBMIT initialization parameter 128
- STAT.SWITCH.TIME initialization parameter 128, 133
- statistics
 - configuration 133
 - display statistics archive file directory 21
 - display status of log 20
 - enabling/disabling statistics for specific record types 25
 - monitoring 136
 - notification of statistics file archival 26
 - statistics initialization failure
 - archive file directory 130
 - archiving 128

- archiving utilities 130
 - DMSTARBT archiving sample 132
 - DMSTARRT archiving sample 131
 - file pairs 127, 128
 - initialization failure 139
 - key sequenced cluster 132
 - sample archiving setup 133
 - SELECT STATISTICS command 128
 - sequence order 127
 - switching between files 128
 - switching the statistics file pair 24
 - STATISTICS ARCHIVED API command 129
 - STATISTICS ARCHIVED command 26
 - Batch Interface usage 27
 - parameters 26, 27
 - statistics exit 95
 - statistics records 97
 - statistics facility
 - archiving 128
 - overview 127
 - STATISTICS ON/OFF command 25
 - Batch Interface usage 25
 - parameters 25
 - STATISTICS SWITCH command 24
 - Batch Interface usage 24
 - parameters 24
 - STATUS field 187
 - Step parameter 30
 - STOP CD command
 - Batch Interface usage 30
 - format 29
 - parameters 29
 - purpose 29
 - STOPCD parameter
 - with Insert User 50
 - with Update User 55
 - STORAGE.SUBPOOL entry field 173
 - SUBMIT command 100
 - submit exit 95, 100
 - control block format 101
 - processing flow 100
 - SUBMIT parameter
 - with INSERT USER 50
 - with UPDATE USER 55
 - SUBMIT PROCESS (SB) menu option 173, 175
 - SUBMIT.EXIT initialization parameter 100
 - SUBMITTER.CMDS parameter
 - with INSERT USER 50
 - with UPDATE USER 55
 - subtask management, interface 162
 - superuser password 65
 - SUPERUSR ID 65
 - SUPPRESS CONNECTION AT STARTUP entry
 - field 178
 - SYNTAX
 - check Network Map control statement 82
 - command example 85
 - system fields 157
- ## T
- T (FREE USER AND TERMINATE USER'S SIGNON)
 - line command 187
 - Tandem LU0 adjacent node example 77
 - TANDEM node type 181
 - Tandem node type 179
 - Task Control Area (TCA) 13
 - TASK parameter 15
 - tasks
 - displaying 16
 - removing from execution 15
 - TCA (Task Control Area) 13
 - TCP/IP
 - adjacent node example 73
 - port number 68
 - TCQ (transmission control queue)
 - description 153
 - file 168
 - TCQ parameter 65
 - TCQE layout 101, 105
 - TCQE trace 28
 - TCQSH trace 28
 - TD CTR field 185, 188
 - TD EXIT COUNT field 195

Index

time field 157
TITLE 206
TOP SECRET 87
TPCB/SYMBOLICS trace 28
Trace files 168
traces
 function
 API session 28
 copy routine 28
 DD names 28
 debug settings 28
 DMGCBSUB 28
 GETMAIN/FREEMAIN 28
 I/O buffer 28
 logon exits 28
 logon processor 28
 pseudo DLBL 28
 RPL 28
 SCIP exit 28
 separate trace per task 28
 session manager 28
 TCQE 28
 TCQSH 28
 TPCB/SYMBOLICS 28
 VM dynamic allocation 28
 VM tape handling 28
 WTO 28
transactions
 CDA 159
 CSSN 159
transient data counter (TD) 186, 188
translation tables
 customization 204
 defaults
 Chinese 5550 204
 Chinese Big5 204
 DBCS-PC Korean 204
 KS5601 204
Transmission Control Queue (TCQ)
 description 153
 file 168
 parameter 65
TS MAX LRECL field 195
TS QUEUE NAME field 195
type commands 33

type defaults file
 overriding attributes 31
 purpose 31
type keys
 binary 33
 binary (/B) 32
 data file (/DF) 32
 data file (DF) 32
 data file 2 (/DF2) 32
 data file 2 (DF2) 32
 MS-DOS communications 32
 OS/2 communications 32
 text 32
TYPE parameter 25
TYPEKEY parameter
 with Delete Type command 42
 with Select Type command 44
 with Update Type command 38

U

U (USER STATUS) option 160
UICB 142
UNIT parameter 36
UNIX LU6.2 adjacent node example 78
UNIX node type 180, 181
UNIX TCP/IP adjacent node example 77
UPDATE
 adjacent node 82
 command example 83
UPDATE TYPE command
 Batch Interface usage 41
 command format 38
 information 31, 37
 optional parameters 38
 purpose 31, 37
 required parameters 38
UPDATE USER command
 authorization record parameters 53
 ADD TYPE 53
 ADD USER 53
 ALTER TYPE 53
 ALTER USER 53
 READ TYPE 53
 READ USER 53

- REMOVE TYPE 53
- REMOVE USER 53
- command format 51
- functional authorization parameters 54
 - CHange 54
 - COPY 54
 - DELPR 54
 - FLUSH 54
 - MAXSA 54
 - MODALS 54
 - MODIFY 54
 - NAME 54
 - PASSword 54
 - PHone 54
 - RESETSA 54
 - RUNJOB 55
 - RUNTASK 55
 - SECURITY 55
 - SELNET 55
 - SELPR 55
 - SELSTAT 55
 - STOPCD 55
 - SUBMIT 55
 - SUBMITTER.COMDS 55
- optional parameters 53
- purpose 51
- required parameters 53
- using batch interface 55
- updating the CONTROL RECORD 176
- UPPER CASE PRINT entry field 191, 193
- user application program, writing 142
- user authorized functions 45
- user commands 45
- user exits 95
- user interface control block 142, 146
- USER PROFILE (SIGNON DEFAULTS) datasets 169
- user sign-on record
 - adding 193
 - deleting 194
 - editing 194
 - reading 194
- USER STATUS (U) option 160
- user status screen, Connect:Direct for CICS
 - Administration 156
- user status, monitoring 188

- USERID field 185, 187
- USERID parameter
 - with DELETE USER 57
 - with INSERT USER 48
 - with SELECT USER 58
- USERID, CICS 160, 192, 194

V

- VER field 163
- VERIFY
 - command example 86
 - node definitions 82
- viewing the work queue display 186
- virtual telecommunications access method (VTAM)
 - domain 161
 - session 163
- VM node type 179, 181
- VM SNA LU0 adjacent node example 78
- VM traces
 - dynamic allocation 28
 - tape handling 28
- VMS node type 179, 181
- VOL=SER parameter 36, 41
- VSAMCAT parameter 37
- VSE node type 179, 181
- VSE/ESA adjacent node example 78
- VSE/ESA LU6.2 adjacent node example 79

W

- W (Work queue display) line command 184
- WHERE parameter
 - UPDATE NETMAP 81
 - with Delete Type command 42
 - with FLUSH TASK command 15
 - with Select Type command 44
- Windows NT Node Type 180, 181
- Windows NT TCP/IP adjacent node example 79
- Work queue display (W) line command 184
- work queue display, viewing 186

Index

work queue screen, Connect:Direct for CICS
Administration 156

WORK TASK field 185

WORK.RETRY.INTERVAL entry field 173, 174

worker subtask (WST) 161

WST (worker subtask) 161

WTO trace 28

Y

Year 2000 compliance 22