

Connect:Direct®

Event Services Support User's Guide

Version 4.5

Connect:Direct Event Services Support User's Guide
Version 4.5
First Edition

(c) Copyright 1998-2005 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 Event Services Support	
	Concepts and Components.....	5
	System Interfaces.....	6
	System Advantages.....	6
Chapter 2	Using Event Services Support	
	Connect:Direct Event Services Support Architecture.....	7
	ESS Data Flow Using CICS API.....	8
	Deciding What Event Data to Collect.....	9
	ESS Event Record Examples.....	10
	Using ESS with the CICS API.....	11
	Using ESS with the ESS User Exit.....	11
	System Architecture.....	12
	Using the ESS Exit.....	12
Chapter 3	Issuing Event Services Commands	
	EVENT SERVICES CREATE Command Format.....	15
	Required Parameters.....	15
	Optional Parameters.....	20
	Sample Command.....	20
	EVENT SERVICES START Command Format.....	20
	Required Parameter.....	20
	Sample Command.....	21
	EVENT SERVICES STOP Command Format.....	21
	Required Parameter.....	22
	Optional Parameter.....	22
	EVENT SERVICES DISPLAY Command Format.....	22
	Required Parameters.....	22
	Optional Parameter.....	22
Chapter 4	Using ESS with the CICS API	
	About the CICS API Interface.....	23

Configuring CICS.....	23
Driver Fields.....	24
Using the CICS API Option.....	25
Linking DMQ012.....	26
Using LASTSEQ.....	28
Reading an Event Record from the Transient Data Queue.....	28

Chapter 5 Record Descriptions

Event Record Type Attributes.....	29
Event Record Types.....	30
Event Record Enhancements.....	32

Glossary

Index

About Connect:Direct Event Services Support

Connect:Direct links technologies and moves information between networked systems and computers. It manages high-performance transfers by providing:

- ❖ Automation
- ❖ Reliability
- ❖ Efficient use of resources
- ❖ Application integration
- ❖ Ease of use

Connect:Direct software 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 Event Services Support (ESS) implements an asynchronous event generation facility in Connect:Direct. ESS is designed for use by external management and automated operations applications that require real-time notification of Connect:Direct activities. ESS supports a publish or subscribe protocol with optional guaranteed event delivery and automated recovery.

Customer applications can register interest in receiving event data, and specify the types of event data the application receives through the new event services commands. ESS supports a general user exit point and an optional API for interface communication with user-written CICS automation applications.

This guide assumes knowledge of the applicable operating system, including its applications, network, and environment. If you are not familiar with the operating systems, refer to the appropriate library of manuals.

Concepts and Components

ESS provides real-time delivery of Connect:Direct event records to interested application interfaces. ESS is based on the existing Connect:Direct statistics facility and is designed to:

- ❖ Minimize the server-performance overhead for ESS processing
- ❖ Prevent adverse impact on Connect:Direct server reliability

ESS is a low-level Connect:Direct server technology that enables you to layer multiple APIs. An end-user application interfaces with the API rather than the low-level ESS core technology. This design enables future extension of ESS to support multiple application interface protocols and simultaneous publishing of the same event data to several event consumer applications.

System Interfaces

ESS supports the event services exit and the CICS API interface.

An application can access event data through an ESS exit. A sample event exit (ESSEVX01) is in the sample library, and writes the event data to a predefined data set.

A CICS API initiates event processing by signing on to Connect:Direct through the current CICS API and issuing ESS commands. The CICS API provided with Connect:Direct writes event data to a CICS Transient Data Queue (TDQ). A sample program, DMQ249, shows how to read an event record from the TDQ.

System Advantages

Building on the existing statistics facility has the following advantages:

❖ Event Definition

Because ESS event records are Connect:Direct statistics records, ESS does not restrict events to abnormal activities. This design enables ESS-enabled applications to initiate procedures based on either successful or unsuccessful Process statements.

❖ Historical Data Recovery

The Connect:Direct statistics file functions as the historical repository of ESS event data. ESS supports optional automated recovery of past event data if an event consumer application fails. Upon restart, ESS requests old event records from the statistics file.

The API automatically manages the restart point for an ESS-enabled application. This API tracks the last successfully delivered event record to a designated subscriber. Upon restart, the API resynchronizes event processing by issuing a SElect STATistics command to retrieve the historical event data and deliver it to the restarted event consumer.

❖ Event Record Types

The event record types give external applications greater visibility into Connect:Direct activities and improve the reliability and responsiveness of interface applications in a production environment. ESS supports several Connect:Direct event record types, including:

- **Process Initialization (PI)** occurs immediately prior to the execution of the first Process statement in a Process.
- **Step Start (CI, JI, TI)** events occur immediately prior to the execution of a COPY, RUN JOB, or RUN TASK statement.
- **COPY Step I/O Start (CE)** occurs immediately prior to the first COPY step I/O in a Process.
- **Queue Change (QW, QH, QE, QT)** events occur whenever a Process moves from one logical queue to another in the TCQ. The sample initialization parameters provided in the base Connect:Direct install the PARMLIB data set and specify STAT.EXCLUDE=(QE, QH, QT, QW). This specification prevents Connect:Direct from producing the queue change records. If you want queue change records, you must set this parameter appropriately.

See Chapter 5, *Record Descriptions* for more information about Event Record Types.

ESS also supports basic event filtering. When an event consumer application registers interest in receiving events, the application can restrict the types of event data that it wants to receive. These options support the same filtering syntax as the SElect STATistics command. Following are two possible selections:

- A fault management application can request notification of COPY requests that did not complete normally, QH (Queue Hold) events or both. When such an event occurs, the fault management application can then poll Connect:Direct through SElect STATistics or SElect PROCess to investigate the problem further.
- An application can register interest in all events for a small group of mission-critical Processes.

Using Event Services Support

ESS uses three main components to provide real-time information about Connect:Direct activities:

- ❖ Client application
- ❖ API
- ❖ Connect:Direct server

ESS can communicate with client application interfaces through the CICS API or through an ESS user exit.

The ESS architecture described in this chapter provides an overview of the system structure. Event data flows are presented in subsequent diagrams. A summary of each type of interface follows the discussion of system operation.

Connect:Direct Event Services Support Architecture

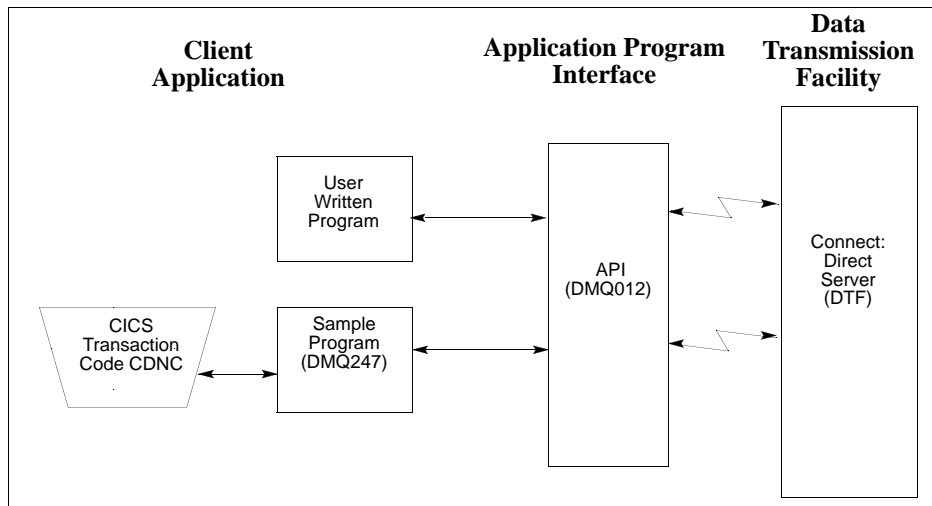
The three main components of ESS interact to accomplish the following tasks:

- ❖ Define which events are reported to the client application
- ❖ Turn event services requests on and off
- ❖ Register event data
- ❖ Report event data

The following diagram details ESS with a CICS application using the Connect:Direct CICS API. The structure consists of the following components:

- ❖ The client application defines the event data requests and issues commands through a user-written program. The DMQ247 sample program demonstrates the method to issue ESS commands.
- ❖ The API receives and passes event data and ESS commands to the Connect:Direct server, and maintains information for the guaranteed delivery feature.

- ❖ The Connect:Direct server is the Data Transmission Facility (DTF) that processes ESS commands and sends the requested event data to the client application through the API (DMQ012).



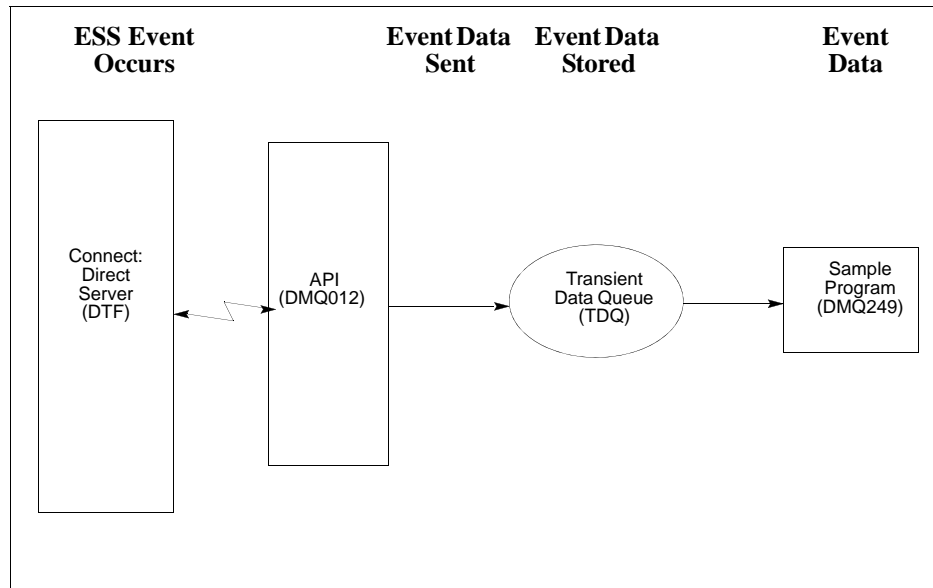
ESS Data Flow Using CICS API

You define the event services environment by specifying the types of event data your application receives. These definitions are the event data requests that you issue through the DMQ247 program or a user-written program.

After you define the event data requests for your client application and activate your request by issuing the `EVENT SERVICES START` command, ESS processes the requested event data as follows:

- ❖ Upon receiving the `EVENT SERVICES START ID` parameter, the API (DMQ012) passes the command to the DTF through a VTAM session.
- ❖ When a requested event occurs, the DTF sends the event data to the API for distribution to the Transient Data Queue (TDQ).
- ❖ The event data is then available to the client application through the DMQ249 Sample Program. The DMQ249 program provides a way to read event data from the TDQ and processes the data for use by your application.

The following diagram illustrates the flow of event data from the DTF to the client application.



For additional information on Event Services commands, refer to Chapter 3, *Issuing Event Services Commands*.

Deciding What Event Data to Collect

Deciding what event data to collect requires that you understand the types of data the ESS delivers and how you can use this data in your event consumer application. Although it is technically possible to request all event data for all Processes in the system, it is usually unnecessary and impractical to do so. Typically, an event consumer application is interested in a subset of event data.

Note: Statistics records excluded through the API STAT command or a DTF statistics exit, or by coding the DTF STAT.EXCLUDE initialization parameter, are not available to ESS.

The following are commonly requested event data:

- ❖ All event data for a particular Process or select group of Processes

Requesting all event data gives the most detailed trace of Process execution. This information is useful during testing of an event consumer application. However, it is typically unnecessary after such an application goes into production.
- ❖ Major event data for a particular Process or select group of Processes

This request is the most common subset specification. For normal production work, this request gives your application a sufficient trace of Process execution, and minimizes overhead for ESS processing and the processing of event records by your application.
- ❖ Subset of event data for a particular Process, set of Processes, or all Processes

For example, an application executes a procedure when a Connect:Direct Process terminates. In this case, it may be sufficient to register interest in PT (Process Termination) records only. As a second example, a fault reporting application needs CT (Copy Termination) statements that did not complete normally (signalling the failure of a Connect:Direct file transfer request).

ESS Event Record Examples

The following examples show the ESS event records from a single Copy Process. The examples show the records produced from Process submittal until normal Process termination. The examples assume that no Connect:Direct commands affecting the Process [including the SElect PROCess command] are issued between Process submittal and termination.

❖ ESS Event Records Example 1 (No Retries)

In this example, the Process does not undergo a Process retry for any reason. The sequence of ESS event records produced by this Process are listed in the following table. These records are in sequential order.

Record Type	Record Description
PS	Process submit
QE	Process moved to EXEC queue
PI	Process initiation
CI	COPY step statement start
CE	COPY step I/O start
CT	COPY termination
NL	Last statement (null step 1)
PT	Process termination

❖ ESS Event Records Example 2 (One Retry)

This example shows the sequence of ESS records that are produced if the same Process undergoes a single Process retry. Connect:Direct commands are deliberately issued to suspend the Process and then release it for retry. As a result, the following record sequence includes the various operator commands required. The events are in sequential order.

Record Type	Record Description
PS	Process submit
QE	Process moved to EXEC queue
PI	Process initiation
CI	COPY step statement start
CE	COPY step I/O start
FS	Suspend Process command issued
CT	COPY termination (as a result of requested SUSPEND)
QH	Process moved to HOLD queue
CH	Change Process command issued with RELEASE=Yes
QE	Process moved to EXEC queue
PI	Process initiation
CI	COPY step start

Record Type	Record Description
CE	COPY I/O start
CT	COPY termination
NL	Last statement in Process end
PT	Process termination

Using ESS with the CICS API

The CICS API writes event data to the Transient Data Queue (TDQ). Your CICS application must include a program to read these event records from the TDQ and pass them to your application for processing.

Follow these steps to enable a CICS application for ESS:

1. Decide what event data your application requires. Refer to the *Deciding What Event Data to Collect* on page 9 for additional information.
2. Verify that the Connect:Direct CICS API software is correctly installed and working. Refer to the Connect:Direct installation documentation for further information.
3. Create an ESS registration program. Include this program in the startup portion of your application. It must be called whenever your application wants to initiate event record collection. This program must perform the following:
 - ♦ Signon to the Connect:Direct CICS API.
 - ♦ Issue the `EVENT SERVICES CREATE` command to register interest in receiving event data and to specify event record filter criteria.
 - ♦ Specify the `ID` parameter to identify a subscriber name for your application. Specify the `WHERE` and `ORWHERE` parameters for your filtering criteria. Refer to the *EVENT SERVICES CREATE Command Format* on page 15 for further information about event filter criteria.
 - ♦ Issue the `EVENT SERVICES START` command and specify the `ID` parameter to initiate event reception.
4. Create a customer interface program to read event records off the CICS TDQ and pass them to your application for processing. The sample interface program (DMQ249) demonstrates how to read an event record from the TDQ. Specifically, your application must issue an `EXEC CICS READQ TD QUEUE` instruction.
5. Modify your application to process the ESS event records.
6. Embed a call to the `EVENT SERVICES STOP ID=subscriber_name` parameter in your application where you want to stop collecting event data.

Using ESS with the ESS User Exit

You can execute Event Services commands through any existing Connect:Direct API such as CICS, ISPF, and DMBATCH. However, not all Connect:Direct APIs can process synchronous event data received from an `EVENT SERVICES START` command. In some cases, it is preferable not to have the Connect:Direct API process synchronous event data. The event services exit provides an alternative. Rather than passing

synchronous event data back to the API that issued the EVENT SERVICES START (ESS) command, an exit processes the event data.

System Architecture

The ESS exit acts as an extension to the Connect:Direct server. When a synchronous event occurs, Connect:Direct calls the event services exit. An event exit control block is passed for each event and contains the following:

- ❖ Address of the event record
- ❖ Flags indicating first or last call
- ❖ Fields for the user to indicate return code and message

DMEVEXCB, found in the sample library, maps the event exit control block. The sample exit, ESSEVX01, is also in the sample library.

The event services exit is called as a subtask from Connect:Direct so that the user code waiting for system services does not directly effect Connect:Direct processing. The sample exit (ESSEVX01) writes each event record to a predefined data set. You must modify ESSEVX01 to specify the name of your event exit data set, and you must define the data set to accommodate records up to 2048 bytes in length.

Using the ESS Exit

Each record passed to the exit consists of a record header and an event record. DMMNHDR, which is in sample library, maps the record header. Members in the sample library also map the event records. See Chapter 5, *Record Descriptions*, to determine which member maps a particular event record.

The exit must reside in a load library accessible to the Connect:Direct server (DTF). Connect:Direct moves event data to storage below the 16M line before calling the event services exit. This step accommodates systems where I/O calls must be done below the line.

To use the ESS user exit, follow these steps:

1. Create an ESS exit program that can process the requested event records.
2. Decide what event data your application requires. Refer to the *Deciding What Event Data to Collect* on page 9 for additional information.
3. Verify that your software is correctly installed and working.
4. Create an ESS registration program. Include this program in the startup portion of your application. It is called whenever your application wants to initiate event record collection. This program must accomplish the following:
 - ◆ Signon to Connect:Direct through the CICS API, the ISPF IUI, or DMBATCH.
 - ◆ Issue the EVENT SERVICES CREATE command to register interest in receiving event data and specify event record filter criteria.

Specify the EXIT parameter to identify the name of your exit program, and the ID parameter to identify a subscriber name for your application. Specify the WHERE and ORWHERE parameters for your filtering criteria. Refer to the *EVENT SERVICES CREATE Command Format* on page 15 for more information about event filter criteria.
 - ◆ Issue the EVENT SERVICES START command and specify the ID parameter to initiate event reception.

5. Embed a call to the `EVENT SERVICES STOP ID=subscriber_name` parameter in your application where you want to stop collecting event data.

If the user application does not process records fast enough, the internal queue, which holds asynchronous events, becomes full. When the queue is full, event processing for this request ends, and Connect:Direct writes a WTOR message. To customize the internal queue size, specify the `MAXQCNT` parameter on the `EVENT SERVICES CREATE` command.

Issuing Event Services Commands

The Event Services commands are the tools that enable you to determine what information your application receives, and to start and stop event notification. You can define the event data through an event services request. This chapter details how to use the EVENT SERVICES commands to create requests, start and stop event notification, and display event data.

This chapter describes the four commands that the Connect:Direct Event Services Support uses:

- ❖ EVENT SERVICES CREATE
- ❖ EVENT SERVICES START
- ❖ EVENT SERVICES STOP
- ❖ EVENT SERVICES DISPLAY

Each command description includes the function of the command and the command parameters.

EVENT SERVICES CREATE Command Format

The EVENT SERVICES CREATE command enables you to create a new request for event data and define which data is sent to your application.

The EVENT SERVICES CREATE command uses the following format and parameters.

Command	Parameters
EVENT [SERVICES] CREATE	ID=event-request-name
	WHERE (CCODE PNAME PNUMBER STARTT STOPT USER SNODE TYPE FNAME CASE LASTSEQ MAXQCNT EXIT)
	ORWHERE= (CCODE PNAME PNUMBER USER SNODE TYPE FNAME)

Required Parameters

The EVENT SERVICES CREATE command has two required parameters.

ID=event-request-name

specifies a logical name for the event services request. This name is a text string of 1-16 characters. You can use any printable characters. You cannot use embedded blanks.

WHERE (CCODE = (condition, completion code)

PNAME = name | (list)
PNUMber = number | (list)
STARTT = ([date | day] [,hh:mm:ssXM])
STOPT = ([date | day] [,hh:mm:ssXM])
USER = name | (list)
SNODE = name | (list)
TYPE = id | (list)
FNAME = dsname | (list)
CASE = YES | NO
LASTSEQ=n
MAXQCNT=n
EXIT=exitname)

specifies selection criteria for event records.

You must specify at least one WHERE () subparameter, such as CCODE, PNAME, PNUMBER, or TYPE.

CCODE = (condition, completion code)

specifies selection by completion code.

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

GT greater than
 LT less than
 EQ equal to
 NE not equal to
 GE greater than or equal to
 LE less than or equal to

completion specifies a completion code value ranging from 1 to 2,147,483,647 so the RUN TASK can pass all values.

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

PNAME = name | (list)

specifies selection by Process name. Specify a list of Processes by enclosing them in parentheses. You can use a wild card character (*) at the end of the name.

For example, if you specify PNAME=TEST*, then all records with TEST in the first four characters of the Process name field are selected. Records having TEST, TEST123, and TESTX all satisfy the selection criterion.

name specifies the name of the Process to select.

(list) specifies a list of Process names to select. Enclose the list in parentheses. Separate Processes in the list with commas.

PNUMBER = number | (list)

specifies selection by Process number. To request a list of Processes, enclose them in parentheses. The range is 1-99999.

number specifies the number of the Process to select.

(list) specifies a list of Process numbers to select. Enclose the list in parentheses. Separate Processes in the list with commas.

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

specifies selection by start date and time. Specify STARTT as a date and time prior to the current time. If you set this parameter, the event records retrieved from the statistics file begin with the date and time you specified and continue until the current time is reached. At that point, event records are processed as they occur.

Note: If you specify STARTT and LASTSEQ with the EVENT SERVICES START command, those values replace the values specified for these parameters in the EVENT SERVICES CREATE command. When you use the time specification in conjunction with the LASTSEQ parameter, you must include hours, minutes, seconds, and hundredths of seconds in the format hh:mm:ss.th.

date specifies the starting date from which event records are retrieved from the statistics file. You can specify the date in either Gregorian or Julian format.

If you use a Gregorian date format, set the DATEFORM initialization parameter to the appropriate date format. Otherwise, the date format defaults to the platform date format. You do not need to set the DATEFORM parameter for Julian date format.

The following table shows the acceptable date formats.

Format	DATEFORM Parameter	Date Format
Gregorian	DATEFORM=MDY	mmddy or mmddy mm/dd/yy or mm/dd/yyyy mm.dd.yy or mm.dd.yyyy
Gregorian	DATEFORM=DMY	ddmmy or ddmmyy dd/mm/yy or dd/mm/yyyy dd.mm.yy or dd.mm.yyyy
Gregorian	DATEFORM=YMD	yyymmdd or yyyyymmdd yy/mm/dd or yyyy/mm/dd yy.mm.dd or yyyy.mm.dd
Gregorian	DATEFORM=YDM	yyddmm or yyyyddmm yy/dd/mm or yyyy/dd/mm yy.dd.mm or yyyy.dd.mm
Julian	N/A	yyddd or yyyyddd yy/ddd or yyyy/ddd yy.ddd or yyyy.ddd

Note: If RETAIN=Y, you cannot specify a date in the STARTT parameter of the initialization file.

day specifies the day of the week to select. Valid names include MOnday, TUEsday, WEDnesday, THursday, FRiday, SATurday, and SUnDay. You can also specify YESTER to retrieve event records created since yesterday, or TODAY to retrieve event records created today.

hh:mm:ssXM indicates the start time of day in hours (hh), minutes (mm), and seconds (ss) selected. XM indicates AM or PM. You can use the 24-hour clock or the 12-hour clock. If you use the 24-hour clock, valid times are 00:00-24:00. If you use the 12-hour clock, valid times are 00:00-12:00 and you must indicate AM or PM. For example, 01:00 hours on the 24 hour clock is expressed as 1:00AM on the 12 hour clock. If you do not specify AM or PM, Connect:Direct assumes the 24-hour clock. You do not need to specify minutes and seconds. You can also specify NOON to retrieve event records

starting at noon, or MIDNIGHT to retrieve event records starting at midnight. The default for the time is 00:00:00, the beginning of the day.

If you do not specify the STARTT parameter, event processing begins with the current time.

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

specifies when interest in event data ends.

date specifies the stop date when event processing ends. You can specify the date in either Gregorian or Julian format.

Refer to the STARTT subparameter on page 17 for a discussion of Gregorian and Julian date formats.

If you specify only the date, the time defaults to 24:00:00.

day specifies the day of the week to select. Valid values include MONday, TUESday, WEDnesday, THursday, FRiday, SATurday, and SUNday. You can also specify TODAY.

hh:mm:ssXM indicates the stop time of day in hours (hh), minutes (mm), and seconds (ss) to select. XM indicates AM or PM. You can use the 24-hour clock or the 12-hour clock. If you use the 24-hour clock, valid times are 00:00-24:00. If you use the 12-hour clock, valid times are 00:00-12:00 and you must indicate AM or PM. For example, 01:00 hours on the 24 hour clock is expressed as 1:00AM on the 12 hour clock. If you do not specify AM or PM, Connect:Direct assumes the 24-hour clock.

You do not need to specify minutes and seconds. You can also specify NOON to retrieve event records starting at noon, or MIDNIGHT to retrieve event records starting at midnight. The default time is 24:00:00, the end of the day.

If you do not specify the STOPT parameter, you must issue an EVENT SERVICES STOP command to stop event processing.

USER = name | (list)

limits the selected event records to those written for userids with the specified name. You can specify a list of names by enclosing them in parentheses. You can use wild card characters. For example, if you specify USER = SYSS\$, then records with SYSS\$ in the first four characters of the userid field are selected. Records having SYSS\$BOB, SYSS\$ADM, and SYSS\$0001 all satisfy this selection criterion. Userid names can be up to 64 characters in length and can contain lowercase characters.

name specifies the userid to select.

(list) specifies a list of userids to select. Enclose the list in parentheses. Separate userids in the list with commas.

SNODE = name | (list)

limits the selected event records to those written for Processes where the specified node name acted as SNODE. You can specify a list of names by enclosing them in parentheses. You can use wild card characters. For example, if you specify SNODE=DALLAS\$, then all records with DALLAS in the first six characters of the SNODE field are selected. Records having DALLAS.PROD, DALLAS.TEST, and DALLAS all satisfy this selection criterion. SNODE names can contain lowercase characters.

name specifies the SNODE to select.

(list) specifies a list of SNODES to select. Enclose the list in parentheses. Separate SNODES in the list with commas.

TYPE = id | (list)

specifies the event record types to select. Every event record that Connect:Direct generates has an associated record-type identifier. Each identifier is two characters long and indicates the event or function that generated the record. The identifier also indicates the record format and contents.

id specifies the event record type.

(list) specifies a list of event record types to select. Enclose the list in parentheses. Separate record types in the list with commas.

See Chapter 5, *Record Descriptions* for a list of all event record type identifiers.

FNAME=dsname | (list)

limits the selected event records to those that contain the specified filename. The FNAME subparameter is valid for the following record types: Copy Termination (CT), Run Job (RJ), Start Connect:Direct (SD), and Submit within Process (SW).

name specifies the filename to select.

(list) specifies a list of filenames to select. Enclose the list in parentheses. Separate filenames in the list with commas.

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

CASE=YES | NO

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

YES changes 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 in the session defaults, if nothing is specified.

LASTSEQ=n

specifies the last sequence number (for Restart capability) associated with the date/time stamp. Use it with the STARTT parameter. This information is provided by CICS when you issue a CREATE EVENT SERVICES REQUEST command for an event services request that finished abnormally. For additional information about how to use LASTSEQ, refer to the *Using LASTSEQ* on page 28.

MAXQCNT=n

specifies the maximum number of elements that can reside in the internal event queue for this request. All events that match the WHERE criteria for this request are placed on an internal queue. If events occur faster than the API can process them, the queue can become full. If the queue becomes full, Connect:Direct terminates event services requests. You can modify the queue size using MAXQCNT. Each queue element takes approximately 2K in storage. The default is **100**.

EXIT=exitname

specifies the name of an EVENT SERVICES OPTION exit. The sample exit ESSEVX01 is provided in the sample library. Event data is processed by an exit rather than the calling API, if EXIT is specified. Refer to *Using ESS with the ESS User Exit* on page 11 for information about the user exit.

Optional Parameters

The following parameters are optional for the EVENT SERVICES CREATE command.

ORWHERE(CCODE = (condition, completion code)

PNAME = name | (list)
PNUMBER = number | (list)
USER = name | (list)
SNODE = name | (list)
TYPE = id | (list)
FNAME = dsname | (list)

specifies which statistics records you want to examine.

The subparameters, such as CCODE, PNAME, PNUMBER, and TYPE, are optional, but you must specify at least one.

The parameters for the ORWHERE keyword are defined in the WHERE keyword section beginning on page 15.

Sample Command

The following is an example of the EVENT SERVICES CREATE command.

```
EVENT CREATE ID=PROCESS_TERM WHERE(TYPE=PT)
```

EVENT SERVICES START Command Format

The EVENT SERVICES START command enables you to start a specific notification request for event data. This command provides parameters where you can define what data (date, time, and sequence) is sent to your application.

The EVENT SERVICES START command uses the following format and parameters.

Command	Parameters
EVENT [SERVICES] START	ID=event-request-name
	WHERE=(STARTT LASTSEQ KEEPREC (CICS only) TRACE) (CICS only)

Required Parameter

The EVENT SERVICES START command has one required parameter.

ID=event-request-name

specifies the logical name of the event services request to start. Specify this name in a CREATE EVENT SERVICES command.

Optional Parameters

The following parameter is optional for the `EVENT SERVICES START` command.

WHERE(STARTT|LASTSEQ|KEEPREC|TRACE)

sets restriction on event records meeting the following selection criteria.

STARTT=(**[date|day][,hh:mm:ssXM]**) specifies selection by designated starting date and time. STARTT must be a date and time prior to the current time. If you specify this parameter, the event records retrieved from the statistics file begins with the date and time you specified, and continues until the current time is reached. At that point, event records are processed as they occur.

For a complete explanation of the subparameters, refer to *EVENT SERVICES CREATE Command Format* on page 15.

LASTSEQ=n specifies the last sequence number (for Restart capability) associated with the date/time stamp. Use this parameter with the STARTT parameter. This information is provided by CICS when you issue a `CREATE EVENT SERVICES REQUEST` command for an event services request that finished abnormally. For additional information about how to use LASTSEQ, refer to *Using LASTSEQ* on page 28.

KEEPREC (CICS-only) specifies that the restart record created by Connect:Direct is kept, regardless of whether the event request completed abnormally or not. If KEEPREC is not specified, the restart record created by Connect:Direct is deleted if the event request completes normally. Refer to *Using LASTSEQ* section on page 28 for more information about how to use the LASTSEQ parameter.

TRACE (CICS-only) specifies that every event record placed on the CICS TDQ is also written to the NDMTRACE temporary storage (TS) queue. You can use the NDMTRACE temporary storage queue to verify that EVENT records are flowing to the transient data queue.

Sample Command

The following is an example of the `EVENT SERVICES START` command.

```
EVENT START ID=PROCESS_TERM
```

EVENT SERVICES STOP Command Format

The `EVENT SERVICES STOP` command enables you to stop event notification for a specific event services request.

Note: The `EVENT SERVICES STOP` command stops event notification and deletes the event request from the system.

The `EVENT SERVICES STOP` command uses the following format and parameters.

Command	Parameters
EVENT [SERVICES] STOP	ID=event-request-name
	IMMEDIATE

Required Parameter

The EVENT SERVICES STOP command has one required parameter.

ID=event-request-name

specifies the logical name of the event services request to start. This name must match a name specified in a CREATE EVENT SERVICES command.

Optional Parameter

The EVENT SERVICES STOP command has one optional parameter.

IMMEDIATE

indicates that no more event records can be sent. If you do not specify IMMEDIATE, all records on the internal EVENT queue are sent before the EVENT request is terminated.

EVENT SERVICES DISPLAY Command Format

The EVENT SERVICES DISPLAY command enables you to display event data for a specific event services request. The EVENT SERVICES DISPLAY command uses the following format and parameter.

Command	Parameter
EVENT [SERVICES] DISPLAY	ID=event-request-name

Required Parameters

The EVENT SERVICES DISPLAY command has no required parameters.

Optional Parameter

The EVENT SERVICES DISPLAY command has one optional parameter.

ID=event-request-name

specifies the logical name of the event services request to display. This name must match a name specified in a CREATE EVENT SERVICES command. If you do not specify this parameter, all event services requests are displayed.

Using ESS with the CICS API

The Connect:Direct CICS interface enables you to use Connect:Direct through the Customer Information Control System (CICS) from local and remote sites. The system includes a set of nested menus, prompts for required information, online Help facilities, and monitoring features for current status.

About the CICS API Interface

The Connect:Direct CICS interface provides 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 Connect:Direct activity in a user-friendly environment.
- ❖ **API** – enables the IUI and user applications to communicate with the Data Transmission Facility (DTF) through the session manager. The API interprets the commands, but the session manager establishes communication sessions and performs standard session management functions.
- ❖ **Connect:Direct DTF** – controls information distribution to other Connect:Direct nodes in the network.

Configuring CICS

You can issue standard Connect:Direct commands from a CICS application program. You can use this interface for both terminal and nonterminal tasks. Following are two typical uses for this API:

- ❖ For terminal tasks, the API enables an installer to provide their own user interface to all or part of Connect:Direct, or work with Connect:Direct from application programs.
- ❖ For non-terminal tasks, the API enables an installer to write background transactions which programmatically issue Connect:Direct commands. The most typical sequence generates a SIGNON TYPE=CICS, and the SIGNOFF results in a cleanup operation of the signon table entry, temporary storage, and so forth.

To avoid signing on before each command and signing off after each command, the user application can pass a logical task number through the Q012TASK field. If the program uses the Q012 TASK, the user program can sign on once and issue multiple commands in conversational mode. A signoff is only needed when the user application is terminating.

An example of the use of this facility is provided in member DMQ247 of the NDMCICS.SAMPLIB. This sample program enables you to type Connect:Direct commands on a screen and view the resulting return code, message number, message text, and Process number assigned to your Connect:Direct Process.

In addition, the CICS application displays the name of the CICS Temporary Storage (TS) queue where the results of your command are stored, the count of items in the queue, and the maximum record length in the queue.

You must install both Connect:Direct and the Connect:Direct CICS interface, and they must be operating for the Connect:Direct CICS API to function.

The following components are required to use DMQ247:

Component	Description
DMQ247	The program source, written in assembly language
DMQM98	The BMS map used by DMQ247
DMQBMSTB	A parsing macro used by DMQ247
DMQCA012	A command-level COMMAREA passed to the Connect:Direct CICS interface

A sample Connect:Direct for CICS API DRIVER screen follows.

```

                                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
EVENT RESTART DATA

PF keys:   3 Exit   5 Signon   6 Command   7 Signoff

```

Driver Fields

The following table describes the driver fields.

Field	Description
COMMAND	This 3-line 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.

Field	Description
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 is 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 of the TS queue in bytes.
EVENT RESTART DATA	This 62-character field contains the DATE/TIME/SEQ of the last event record successfully received. Data is only displayed if an EVENT SERVICES CREATE command is issued and a previous event services request ended abnormally.

Using the CICS API Option

To use the sample program, you must first change DMQ247 to issue a valid SIGNON command for your environment. (The SIGNON command is defined near the end of the source module.) The command must include a valid USERID, PASSWORD, and NODE. Before executing this program as a transaction, you must be signed on to CICS, and then assemble DMQM98 and DMQ247. Sample JCL is in the NDMCICS.SAMPLIB. Assemble DMQM98 and DMQ247 as follows:

1. Use member ASMBMS to assemble DMQM98.
2. Use member ASMCICS to assemble DMQ247.

Note: DMQ247 must be link-edited as NORENT.

Although you can use any transaction code to invoke DMQ247, transaction CDNC is already defined to execute program DMQ247. Processing for DMQ247 proceeds as follows:

- ❖ Use the Connect:Direct Administrative (CDA) transaction to verify that the Connect:Direct CICS interface and the requested node are both active.
- ❖ The program checks for the presence of a communications area in the Exec Interface Block (EIB). If no communications area is present, or if you press the **Clear** key, the DMQM98 map is sent and a RETURN TRANSID is performed to invoke the transaction again when the **Enter** key is pressed.
Note: If you press the **PF3** or **PF15** key, the program terminates.
- ❖ The DMQM98 map is received. If you press **PF5** or **PF17**, 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 displayed on DMQM98 after the SIGNON is zero), type a valid Connect:Direct command on the line provided in the DMQM98 map. Press **PF6** or **PF18** to send the command to the Connect:Direct DTF. Command results are displayed when they are returned from the DTF.
- ❖ To sign off from the DTF, press **PF7** or **PF19**.

When you type a command through the DMQM98 screen, its length is determined and the address of the length and command are placed in the DMQCA012 communications area at label Q012CMDA. Program DMQ012 is

then invoked through an EXEC CICS LINK command. When control returns to DMQ247, the DMQCA012 communications area contain the results of the command.

Note: The DMQ247 does not display the results of the issued command that are stored in CICS temporary storage. You can retrieve these records programmatically or view them using the CICS CEBR transaction.

Use the techniques in the DMQ247 sample program to issue a valid Connect:Direct command. Results of commands such as SELECT PROCESS and SELECT STATISTICS are written to CICS temporary storage. Other commands may produce no output.

Linking DMQ012

Access the API by linking program DMQ012 as follows.

```
EXEC CICS LINK PROGRAM('DMQ012') COMMAREA(Q012COMM)          +
      LENGTH(Q012CMLH)
```

The COMMAREA Q012COMM is defined by macro DMQCA012 and is provided in the sample library. A COBOL version of this record layout is provided as member DMQAPIC in the sample library. A brief description of each of the fields follows.

Q012CMDA

DS XL4

Full word containing the address of the command to issue to Connect:Direct. The command must be in the following format.

CMDLEN DS H

Length of CMDTEXT that Connect:Direct recognizes, not including length of CMDLEN.

CMDTEXT DS Clx

Command text

Q012RETC

DS XL4

Return code received after the command is invoked.

Return Code	Description
0	request completed successfully
1	C:D-CICS interface level error
2	C:D-CICS node level error
3	C:D-CICS signon level error
5	invalid COMMAREA or command passed to CICS API
8 and higher	unsuccessful request

Q012TDCT

DS XL4

TDEXIT ITEM COUNT. The field contains the number of records received from the Connect:Direct DTF.

Q012TDMX

DS XL2

TDEXIT MAX ITEM SIZE – the maximum record size received from the Connect:Direct DTF.

Q012PROC

DS CL6

Process number of the Process just submitted.

Q012TSKY

DS CL8

Temporary storage ID – the name of the CICS TS queue that contains the command output.

Q012MSID

DS CL8

Message ID returned from the Connect:Direct DTF.

Q012MSTX

DS CL64

Message Text returned from the Connect:Direct DTF.

Q012DTE

DS PL4

Date of the last event acknowledgment.

Q012TME

DS XL4

Time of the last event acknowledgment in the format hh:mm:ss:th where hh is hours, mm is minutes, ss is seconds, and th is hundredths of a second.

Q012SEQ

DS XL2

Sequence number of the last event acknowledgment.

Q012TASK

DS CL4

Logical task number to be used by Connect:Direct for CICS for nonterminal tasks.

Q012CMDA is the only required parameter you must set before linking to DMQ012. For optimum performance, follow the example of DMQ247 and initialize all fields before linking to DMQ012.

Any output generated by issuing the command is returned in a temporary storage queue. Your terminal ID is displayed as the last four characters in the unique TS queue name.

Using LASTSEQ

All event records contain a unique timestamp that consists of date, time, and sequence number. Each EVENT SERVICES request is associated with an ID. The Connect:Direct CICS API creates a restart record for an ID when an EVENT SERVICES CREATE command is processed. After an event record is successfully delivered to the Transient Data Queue, its timestamp is saved in the restart file. If event services processing ends abnormally, the restart record is retained. The restart record is also retained when you specify the KEEPRREC (keep restart record) in the EVENT START statement.

When an EVENT SERVICES CREATE command is processed, the restart file is searched for a record matching the ID of the CREATE. If a match is found, the date, time, and sequence number of the last successfully delivered event record is returned. When you use the time specification in conjunction with the LASTSEQ parameter, you must include hours, minutes, seconds, and hundredths of seconds in the format hh:mm:ss.th.

If missed event data is required, you can use this information as the STARTT and LASTSEQ values as in the following example.

```
EVENT SERVICES START ID=id
WHERE (STARTT=(date,time) LASTSEQ=seq)
```

Event records beginning after this timestamp are processed.

Reading an Event Record from the Transient Data Queue

The sample program DMQ249 reads an ESS event record from the Transient Data queue. The sample program DMQ249 is in \$CD.SAMPLIB. You can use sample JCL ASMCICS to assemble DMQ249.

Each record passed to the user application consists of a record header and an event record. DMMNHDR, which is in the Connect:Direct sample library, maps the record header. Members in the sample library also map the event records. See Chapter 5, *Record Descriptions*, to determine which member maps a particular event record.

Record Descriptions

ESS enables you to create ESS-enabled applications. This chapter provides reference information about event record types and attributes that help you create applications. For a list of statistics record types, refer to Chapter 10, *Using Connect:Direct Exits*, in the *Connect:Direct OS/390 Administration Guide*.

Assembler DSECT members are in the sample library. Access the information by the member names listed in the following table. Use the two-character designations for record types when you browse the Connect:Direct statistics file.

Event Record Type Attributes

ESS has several statistics record types. This section summarizes the attributes of these record types.

Most record types have the following common attributes:

- ❖ Process name
- ❖ Process number
- ❖ Message ID
- ❖ Submitter's symbolic node name
- ❖ PNODE name for this Process
- ❖ SNODE name for this Process
- ❖ THISNODE (This node is P[node] or S[node])

The following paragraphs detail the attributes of specific record types. For ease of reference, the information is grouped according to event type or enhancement type, and they are presented alphabetically.

Event Record Types

Record type attributes for event records are described in the following paragraphs. The information is organized alphabetically by record type identifier.

CI (COPY Step Initiation)

This record is created immediately prior to the execution of any tasks associated with a COPY statement. The intent of this record is to capture the timing of such work as data set allocation prior to any I/O occurring. In addition to the common attributes, CI has the following individual attributes:

- ❖ COPY step start date/time
- ❖ Step name or label
- ❖ CI record retry count
- ❖ Transfer direction
- ❖ Source file name
- ❖ Destination file name
- ❖ Member name (when applicable)

CE (COPY I/O Start)

This record is created immediately prior to the first block of data transferred in a COPY statement. In addition to the common attributes, CE has the following individual attributes:

- ❖ COPY I/O start date/time
- ❖ Step name or label
- ❖ CE record retry count

EI (Event Request Initiation)

This record is created upon completion of a EVENT SERVICES START command.

Note: The common attributes for Process-related records do not apply to this record.

The following attributes apply to this event:

- ❖ Date/Time the command completed
- ❖ User ID of the user who issued the command
- ❖ Event services request ID
- ❖ Completion code for the command
- ❖ Command parameter string

EL (Event Session Lost)

This record is created when the Connect:Direct DTF is lost.

ET (Event Request Stop)

This record is created when an EVENT SERVICES STOP command completes.

Note: The common attributes for Process-related records do not apply to this record.

The following attributes apply to this event:

- ❖ Date/Time the command completed
- ❖ User ID of user who issued the command
- ❖ Event services request ID
- ❖ Completion code for the command

JI (RUN JOB Start)

This record is created immediately before the job specified by the RUN TASK statement is submitted. In addition to the common attributes, JI has the following individual attributes:

- ❖ RUN JOB start date/time
- ❖ Step name or label
- ❖ Job name
- ❖ JI record retry count

QE (Process moved to EXEC queue)

This record is created when a Process is moved to the execution queue. In addition to the common attributes, QE has the following individual attributes:

- ❖ Process execution start date/time
- ❖ QE record retry count

PI (Process Initiation)

This record is created immediately prior to when the first Process step is executed. In addition to the common attributes, PI has the following individual attributes:

- ❖ Queue change date/time
- ❖ Execution queue status value
- ❖ PI record retry count

QH (Process moved to the HOLD queue)

This record is created whenever a Process is moved to the Hold queue, either through some problem during Process execution or submission, a SUSPEND PROCESS command issued or a session failure. The reason for moving to the Hold queue is generally identified by the Hold queue status value. In addition to the common attributes, QH has the following individual attributes:

- ❖ Queue change date/time
- ❖ Hold queue status value
- ❖ QH record retry count

QT (Process moved to the TIMER queue)

This record is created whenever a Process is moved to the Timer queue, normally through a Process retry. In addition to the common attributes, QT has the following individual attributes:

- ❖ Queue change date/time
- ❖ Timer queue status value
- ❖ QT record retry count

QW (Process moved to the WAIT queue)

This record is created whenever a Process is moved to the Wait queue from the Hold or Timer queues. In addition to the common attributes, QW has the following individual attributes:

- ❖ Queue change date/time
- ❖ Wait queue status value
- ❖ QW record retry count

TI (RUN TASK Initiation)

This record is created immediately prior to the initiation of the task specified by the RUN TASK statement. In addition to the common attributes, TI has the following individual attributes:

- ❖ RUN TASK start date/time
- ❖ Step name or label
- ❖ Program name
- ❖ TI record retry count
- ❖ Date/time
- ❖ Process name and number
- ❖ Event name
- ❖ Event trigger
- ❖ Event type

Event Record Enhancements

An enhancement to existing statistics records is described in the following paragraph.

CT (COPY Termination) Record Enhancements

The existing CT record is enhanced to include the following information:

- ❖ Total number of retries for the COPY step
- ❖ Total amount of data moved for all attempts of the COPY

A

ABEND

A task that ends prematurely, or abnormally, due to an error that cannot be resolved by recovery facilities while the task is executing.

ACB

See *Access Method Control Block (ACB)*.

Access Method

A technique for moving data between main storage and input/output (I/O) devices.

Access Method Control Block (ACB)

A control block that links an application program to VSAM or VTAM.

Adapter

A hardware card that allows a device, such as a PC, to communicate with another device, such as a monitor, a printer, or other I/O device. See also Channel-to-Channel Adapter.

ADJACENT.NODE

An entry in the network map. Adjacent node entries define nodes in the network with which the local Connect:Direct may communicate. Each entry specifies a locally used Connect:Direct name, its associated network communications name, and session control parameters for these nodes.

API Pool

Identifies the APPLIDs to be used for API communication with the DTF.

Application Program Interface (API)

The Connect:Direct component that accepts commands from the Interactive User Interface (IUI), Batch Interface, the Operator Interface, or user-written program and places them in a format so that the user's request can be executed by the DTF. If there are errors, the API returns a message to the user. If there are no errors, the API sends the command to the DTF using a VTAM session.

APPLID

The name specified in the ACB macro that identifies the application program to VTAM. For Connect:Direct, these APPLIDs correspond to a DTF node name or an API APPLIDs.

Asynchronous Processes

Processes that occur without a regular or scheduled time relationship. Unexpected or unpredictable with respect to the instructions of the program or to time. Contrast with synchronous.

Attributes

Characteristics or properties that can be controlled, usually to obtain a required appearance; for example, the color of a line.

Authorization File

Connect:Direct file used to control access to Connect:Direct and identify commands that can be executed by user ID. This file can also be used in conjunction with security exit interfaces to support the secured point-of-entry feature.

B

Batch Interface

An interface where non-interactive programs are executed. The environment schedules their execution independently of their submitter. Connect:Direct users issue batch commands using DMBATCH, a Connect:Direct-supplied program.

Buffer

1. A portion of storage used to hold input or output data temporarily.
2. A routine or storage used to compensate for a difference in data rate or time of occurrence of events, when transferring data from one device to another.

Buffer Pool

A set of buffers that contains buffers of the same length.

C

Central Processing Unit (CPU)

The part of a computer that includes the circuits that control the interpretation and execution of instructions.

CF

See *Coupling Facility (CF)*.

Channel

1. A functional unit, controlled by an S/390 server that handles the transfer of data between processor storage and local peripheral equipment.
2. A path along which signals can be sent.
3. The portion of a storage medium that is accessible to a given reading or writing station.
4. In broadband transmission, a designation of a frequency band 6 MHz wide.

Channel-to-Channel (CTC)

Refers to the communication (transfer of data) between programs on opposite sides of a channel-to-channel adapter (CTCA). The CTCA for Connect:Direct OS/390 can be an ESCON CTC.

Channel-to-Channel Adapter (CTCA)

A hardware device that can be used to connect two channels on the same computing system or on different systems. The CTCA for Connect:Direct OS/390 can be an ESCON CTC.

Checkpoint/Restart

Eliminates the need to retransmit an entire file in the event of a transmission failure. A value on the COPY statement or in the initialization parameter, CKPT.MODE and CKPT, specifies the checkpoint interval. If a copy procedure is interrupted, Connect:Direct will restart that copy at the last checkpoint.

CICS

See *Customer Information Control System (CICS)*.

Command Line Interface

Connect:Direct interface that allows users to submit Connect:Direct Processes and commands from their native command line environment.

Commands

An instruction that directs a control unit or device to perform an operation or a set of operations. Connect:Direct users issue commands to initiate and monitor activity within the Connect:Direct system. Connect:Direct commands can be issued from the IUI, the operator console, a batch job, or a user application program.

Component

1. Hardware or software that is part of a functional unit.
2. A functional part of an operating system; for example, the scheduler or the Hold queue.

Compression

Storing data in a format that requires less space than usual. Data compression is particularly useful in communications because it enables devices to transmit the same amount of data in fewer bits. See also Variable Extended Compression.

Configuration

The arrangement of a computer system or network as defined by the nature, number, and main characteristics of its functional units. More specifically, the term configuration may refer to a hardware or software configuration. See also System Configuration.

Connect:Direct/Manager

The component of a Connect:Direct/Plex environment that handles the following functions:

- ❖ Interface connections
- ❖ Statistics file updates
- ❖ CKPT and TCQ/TCX file access
- ❖ TYPE file, AUTH file, NETMAP file, SECURE+ parameter file, and SECURE+ Digital Signature updates
- ❖ Workload balancing

Connect:Direct/Plex

A Connect:Direct system consisting of a Connect:Direct/Manager and one or more Connect:Direct/Servers in a system complex or parallel system complex.

Connect:Direct/Server

A Connect:Direct/Plex component that executes the Processes.

Connect:Direct/Stand-alone Server

A Connect:Direct system that is not part of a Connect:Direct/Plex.

Connectivity

A term used to describe the physical interconnections of multiple devices, computers, or networks employing similar or different technology and/or architecture together to accomplish effective communication between and among connected members. It involves data exchange and/or resource sharing.

Console

A logical device that is used for communication between the user and the system.

Coupling Facility (CF)

A special logical partition (LP) that provides high-speed caching, list processing, and locking functions in Parallel Sysplex.

CPU

See *Central Processing Unit (CPU)*.

Cross-System Coupling Facility (XCF)

An OS/390 facility that allows multiple instances of the same application to communicate and share information with each other.

CTC

See *Channel-to-Channel (CTC)*.

CTCA

See *Channel-to-Channel Adapter (CTCA)*.

Customer Application

An application that does customer-specific processing.

Customer Information Control System (CICS)

An IBM-licensed program that enables transactions entered at remote terminals to be processed concurrently by user-written application programs. It includes facilities for building, using, and maintaining databases.

D**Daemon**

A background task, process, or thread that intermittently awakens to perform some task or function and then returns to an idle state.

DASD

See *Direct Access Storage Device (DASD)*.

Database

1. A set of data, or a part or the whole of another set of data, that consists of at least one file and is sufficient for a given purpose or for a given data-processing system.
2. A collection of data fundamental to a system. See also Database Control (DBCTL), data entry database (DEDB), data sharing, and data sharing group.

Data Set

The major unit of data storage and retrieval, consisting of a collection of data in one of several prescribed arrangements and described by control information to which the system has access.

Data Sharing

In a Parallel Sysplex, the ability of concurrent subsystems or application programs to directly access and change the same data while maintaining data integrity.

Data Transmission Facility (DTF)

The nucleus component of Connect:Direct. The DTF controls information distribution to other Connect:Direct nodes in the network. Startup parameters that govern the overall activity of the DTF are defined within the initialization parameters.

In a Connect:Direct/Plex, the DTF consists of a Connect:Direct Manager and one or more Connect:Direct Servers.

Decipher

To convert enciphered data into clear data.

Decrypt

To convert encrypted data into clear data.

Default

Pertaining to an attribute, value, or option that is assumed when none is explicitly specified.

Direct Access Storage Device (DASD)

A physical device, such as an IBM 3390, in which data can be permanently stored and subsequently retrieved using licensed products like IMS and DB2, or using IBM supported access methods like VSAM in operating system environments like OS/390.

Directory

A list of files that are stored on a disk or diskette. A directory also contains information about the file, such as size and date of last change.

DTF

See *Data Transmission Facility (DTF)*.

Dynamic

Pertaining to an operation that occurs at the time it is needed rather than at a predetermined or fixed time.

E**Execution**

The process by which a computer carries out the instruction or instructions of a computer program.

Extended Submit Facility (ESF)

The facility that allows users to queue data transfer requests to a Connect:Direct node that is not active. This allows users to submit work to Connect:Direct, even if the Connect:Direct DTF is down.

F

File System

The collection of files and file management structures on a physical or logical mass storage device such as a disk.

Format

1. A specified arrangement of things, such as characters, fields, and lines, usually used for displays, printouts, or files.
2. To arrange things such as characters, fields, and lines.

H

Hardware

The physical equipment as opposed to programs, procedures, rules, and associated documentation. Contrast with software.

Host (computer)

1. In a computer network, a computer that provides end users with services such as computation and databases and that usually performs network control functions.
2. The primary or controlling computer in a multiple-computer installation.

I

ICO

See *InterConnect Option (ICO)*.

Initialization

The preparation of a system, device, or program for operation. Connect:Direct initialization parameters specify alternate values for various parameters used during Connect:Direct start up.

Input/Output (I/O)

1. Pertaining to a device whose parts can perform an input process and an output process at the same time.
2. Pertaining to a functional unit or channel involved in an input process, output process, or both, concurrently or not, and to the data involved in such a process.

Interactive

Pertaining to a program or system that alternately accepts input and then responds. An interactive system is conversational; that is, a continuous dialog exists between user and system. Contrast with batch.

Interactive User Interface (IUI)

An ISPF screen and dialog component that allows users to define and submit Connect:Direct Processes as well as issue Connect:Direct commands that monitor and control administrative and operations activity. An IUI is also available for a CICS environment.

InterConnect Option (ICO)

A feature of Connect:Enterprise. This option provides an automatic, secure way to route application-produced distribution files from a Connect:Direct supported node to a Connect:Enterprise node for distribution, automatically distribute Connect:Enterprise batches to a Connect:Direct node upon arrival, and provide success or failure notification at each process step.

Interface

A shared boundary. An interface might be a hardware component to link two devices or it might be a portion of storage or registers accessed by two or more computer programs.

Interrupt

1. A suspension of a process, such as execution of a computer program caused by an external event, and performed in such a way that the process can be resumed.
2. In data communication, to take an action at a receiving station that causes the sending station to end a transmission.
3. To temporarily stop a process.

I/O

See *Input/Output (I/O)*.

I/O Service Units

A measure of individual data set I/O activity and JES spool reads and writes for all data sets associated with an address space.

IUI

See *Interactive User Interface (IUI)*.

J

Job Entry Subsystem (JES)

A system facility for spooling, job queuing, and managing job-related data.

L

LAN

See *Local Area Network (LAN)*.

Link

The combination of physical media, protocols, and programming that connects devices.

Load Module

A computer program in a form suitable for loading into storage for execution.

Local Area Network (LAN)

A data network located on the user's premises in which serial transmission is used for direct data communication among data stations. It services a facility without the use of common carrier facilities.

Local Cache

A buffer in local system storage that may contain copies of data entries in a CF cache structure.

LOCAL.NODE

An entry in the Network Map. The local node entry defines the logical Connect:Direct name of the local Connect:Direct DTF and its associated communications name. The local node entry also contains the name of the transmission queue and the SUPERUSR ID password, if specified.

Logical Connection

In a network, devices that can communicate or work with one another because they share the same protocol.

Logical Unit (LU)

In VTAM, the source and recipient of data transmissions. Data is transmitted from one logical unit (LU) to another LU. For example, a terminal can be an LU, or a CICS system can be an LU.

Logically Partitioned (LPAR) Mode

A CPC power-on reset mode that enables use of the PR/SM (Processor Resource/Systems Manager) feature and allows an operator to allocate CPC hardware resources (including CPs, central storage, expanded storage, and channel paths) among logical partitions.

LU

See *Logical Unit (LU)*.

M

Main Storage

A logical entity that represents the program addressable portion of central storage. All user programs are executed in main storage.

Mainframe (S/390 server)

A large computer, in particular one to which other computers can be connected so that they can share facilities the S/390 server provides, for example, an S/390 computing system to which personal computers are attached so that they can upload and download programs and data.

Memory

The program-addressable storage from which instructions and other data can be loaded directly into registers for subsequent execution or processing.

Migration

Installing a new version or release of a program when an earlier version or release is already in place. See file migration.

Modal Statements

Statements (IF THEN, EIF, ELSE, EXIT, and GOTO) in Connect:Direct that allow you to alter the sequence of Connect:Direct Process execution based on completion of a previous Process step.

Module

A program unit that is discrete and identifiable with respect to compiling, combining with other units, and loading; for example, the input to or output from an assembler, compiler, linkage editor, or executive routine.

Multiprocessing

The simultaneous execution of two or more computer programs or sequences of instructions. See also Parallel Processing.

N

NCP

See *Network Control Program (NCP)*.

Netmap

See *Network Map*.

Network

A configuration of data processing devices and software connected for information interchange.

Network Control Program (NCP)

A program residing in a communication controller (for example, the IBM 3745 Communication Controller) that controls the operation of the communication controller.

Network Map

The VSAM file that identifies all valid Connect:Direct nodes and applids in the network. There is one Network Map (netmap) associated with each Connect:Direct node. There is one entry in that netmap for each of the other Connect:Direct nodes to which the local Connect:Direct node can initiate a session. The netmap entries also contain the rules or protocol to which the nodes will adhere when communicating.

Node

1. Any site in a network from which information distribution can be initiated.
2. In SNA, an endpoint of a link or junction common to two or more links in a network. Nodes can be distributed to S/390 server processors, communication controllers, cluster controllers, or terminals. Nodes can vary in routing and other functional capabilities.

O**Offline**

Not controlled directly by, or not communicating with, a computer. Contrast with online.

Online

Pertaining to equipment, devices, or data under the direct control of the processor. Contrast with offline.

Online Messages

The completion and error messages that are displayed online.

Operating System (OS)

The software that controls the execution of programs and that may provide services such as resource allocation, scheduling, input/output control, and data management. Although operating systems are predominantly software, partial hardware implementations are possible. Examples are OS/390, VSE/ESA, and VM/ESA.

Operator Interface

Allows Connect:Direct commands to be issued from the operator console. This interface also allows tailoring of Connect:Direct commands through a command list (CLIST) facility.

OS

See *Operating System (OS)*.

P

Parallel

1. Pertaining to a process in which all events occur within the same interval of time, each handled by a separate but similar functional unit; for example, the parallel transmission of the bits of a computer word along the lines of an internal bus.
2. Pertaining to the concurrent or simultaneous operation of two or more devices or to concurrent performance of two or more activities in a single device.
3. Pertaining to the concurrent or simultaneous occurrence of two or more related activities in multiple devices or channels.
4. Pertaining to the simultaneity of two or more processes.
5. Pertaining to the simultaneous processing of the individual parts of a whole, such as the bits of a character and the characters of a word, using separate facilities for the various parts.

Parallel Processing

The simultaneous processing of units of work by many servers. The units of work can be either transactions or subdivisions of large units of work (batch).

Parallel Sessions

The capability of having two or more concurrently active sessions between the same set of two LUs. With parallel session support, Connect:Direct allows multiple, concurrent file transfers between two Connect:Direct nodes.

Parallel Sysplex

A sysplex with one or more coupling facilities.

Partitioned Data Set (PDS)

A data set in DASD storage that is divided into partitions, called members, each of which can contain a program, part of a program, or data.

Partitioned Data Set Extended (PDSE)

A data set organization that stores many different but related members. A PDSE contains an indexed directory and members that are similar to the directory and members of a partitioned data set (PDS) but offers architectural advantages that a PDS cannot. For example, space can be dynamically allocated and reclaimed using PDSEs, exploiting space more efficiently.

Primary Node (PNODE)

The Connect:Direct node on which the Process is being submitted. The primary node may also be referred to as the controlling or source node, but should not necessarily be interpreted as the sending node since PNODE can be the receiver. In every Process, there is one PNODE and one SNODE specified. The submitter of a Process is always the PNODE.

Process

A series of statements (which can be predefined and stored in a library) submitted through the API to initiate Connect:Direct activity, such as copying files, running jobs, and so on.

Process Statements

The statements that are used to build a Connect:Direct Process. They contain instructions for transferring files, running operating system jobs, executing programs, or submitting other Connect:Direct Processes. Process statements include COPY, RUN JOB, RUN TASK, SUBMIT, SYMBOL, and modals (conditional logic).

Processing Unit

The part of the system that does the processing, and contains processor storage.

Processor

A processing unit, capable of executing instructions when combined with main storage and channels.

Protocol

A specification of the format and relative timing of information exchanged between peer entities within a layer.

R

Record

A set of data treated as a unit.

Recovery

To maintain or regain system operation after a failure occurs. Generally, to recover from a failure is to identify the failed hardware, to de-configure the failed hardware, and to continue or restart processing.

Remote Node

The Connect:Direct node that interacts with the local node during Process execution. The remote node is also referred to as the participating, receiving, target, destination, or secondary node (SNODE).

Resource Access Control Facility (RACF)

The facility that provides access control by identifying and verifying users to the system. RACF authorizes access to resources, logs unauthorized access attempts, and logs accesses to protected data sets.

Response Time

The amount of time it takes after a user presses the enter key at the terminal until the reply appears at the terminal.

Retry Interval

An installation parameter that specifies the interval, in minutes, that the retries mentioned in the Max Retries parameter will be performed.

Routing

The assignment of a path by which a transfer reaches its destination.

S

Secondary Node (SNODE)

The Connect:Direct node that interacts with the primary node (PNODE) during process execution. The secondary node (SNODE) can also be referred to as the participating, target, or destination node. Every Process has one PNODE and one SNODE.

Session

1. The entity through which a Connect:Direct PNODE initiates and executes one or more Processes to one or more SNODEs.
2. The entity through which a Connect:Direct SNODE receives one or more Processes.
3. In SNA, a logical connection between two network addressable units that can be activated, tailored to provide various protocols, and deactivated as requested.
4. The data transport connection resulting from a call or link between two devices.
5. The period of time during which a user of a node can communicate with an interactive system; usually it is the elapsed time between logon and logoff.
6. In network architecture, an association of facilities necessary for establishing, maintaining, and releasing connections for communication between stations.

Session Classes

The installation parameter that specifies the Process class groupings, priorities and number of Processes that can be concurrently executed on this Connect:Direct node.

Shared

Pertaining to the availability of a resource to more than one use at the same time.

SNA

See *Systems Network Architecture (SNA)*.

SNODE

See *Secondary Node (SNODE)*.

SRF

See *Sysplex Requester Facility (SRF)*.

Standby Connect:Direct Manager

In an extended recovery environment, the backup Connect:Direct Manager that takes over work from the active Connect:Direct Manager when the active Connect:Direct Manager fails.

Standby Connect:Direct Server

In an extended recovery environment, the backup Connect:Direct Server that takes over work from the active Connect:Direct/Server when the active Connect:Direct Server fails.

Standby Connect:Direct System

In an extended recovery environment, the backup Connect:Direct system that takes over work from the active Connect:Direct system when the active system fails.

Statistics Facility

The Connect:Direct facility that records all Connect:Direct activities.

Statistics Files

A pair of VSAM data sets that hold Connect:Direct statistics records to document the history of a Connect:Direct Process.

Storage

A unit into which recorded data can be entered, in which it can be retained and processed, and from which it can be retrieved.

Subsystem

A secondary or subordinate system, or programming support, that is usually capable of operating independently of or asynchronously with a controlling system.

SYMBOL Statement

The Connect:Direct Process statement that allows you to build symbolic substitution values.

Symbolics

The parameters that allow one predefined Process to be used for multiple applications. For example, the file names for a COPY operation could be passed to the Process by the user submitting the Process.

Synchronous

1. Pertaining to two or more processes that depend on the occurrences of a specific event such as common timing signal.
2. Occurring with a regular or predictable timing relationship.

Sysplex

A set of OS/390 systems communicating with each other through certain multisystem hardware components and software services to process workloads.

Sysplex Data Sharing

The ability of multiple OS/390 subsystems to share data across multiple system images. Sysplex data sharing differs from two-way data sharing in that the latter allows sharing across only two system images.

Sysplex Requester Facility (SRF)

A component of Connect:Direct for OS/390 that enables interactive, batch, and application access to DTFs in the system. The SRF runs in an LPAR or on a machine where no DTF is installed. See also Parallel Sysplex and Sysplex Data Sharing.

System

In data processing, a collection of people, machines, and methods organized to accomplish a set of specific functions.

System Configuration

A process that specifies the devices and programs that form a particular data processing system.

Systems Network Architecture (SNA)

A network architecture designed to provide compatibility among a wide variety of hardware and software products so that they can be used to build complex networks. It defines protocols, standards, and message formats to which different hardware and software products must conform.

The SNA network consists of network addressable units (NAUs), boundary function components, and the path control network.

T

TCQ

See *Transmission Control Queue (TCQ)*.

TCP/IP

See *Transmission Control Protocol/Internet Protocol (TCP/IP)*.

TDQ

See *Transient Data Queue (TDQ)*.

Terminal

A device that is capable of sending and receiving information over a link; it is usually equipped with a keyboard and some kind of display, such as a screen or a printer.

Throughput

1. A measure of the amount of work performed by a computer system over a given period of time, for example, number of jobs per day.
2. A measure of the amount of information transmitted over a network in a given period of time.

Transient Data Queue (TDQ)

A CICS temporary storage queue in which event data is stored so that a client application can retrieve the information.

Transmission Control Protocol/Internet Protocol (TCP/IP)

A set of public domain networking protocol standards that specify the details of how computers communicate, as well as a set of conventions for interconnecting networks and routing traffic.

Transmission Control Queue (TCQ)

A VSAM relative record data set (RRDS) used to hold all Processes that have been submitted to Connect:Direct.

V

Variable Extended Compression

The variable compression capability supported by Connect:Direct. This compression will be used when extended compression is specified and both sides support variable compression.

Virtual Storage (VS)

The storage space regarded as addressable main storage by the user of a computer system in which virtual addresses are mapped into real addresses. Virtual storage size is limited by the address scheme of the computer system and the amount of auxiliary storage available, rather than the actual number of main storage locations.

Virtual Storage Access Method (VSAM)

An access method for direct or sequential processing of fixed and variable-length records on direct access devices. The records in a VSAM data set or file can be organized in logical sequence by a key field (key sequence), in the physical sequence in which they are written on the data set or file (entry-sequence), or by relative-record number.

Virtual Telecommunication Access Method (VTAM)

The program that provides for workstation and network control. It is the basis of a System Network Architecture (SNA) network. It supports SNA and certain non-SNA terminals. VTAM supports the concurrent execution of multiple telecommunications applications and controls communication among devices in both single processor and multiple processor networks.

W

Wide Area Network (WAN)

A network that provides communication services to a geographic area larger than that served by a local area network.

X

XCF

See *Cross-System Coupling Facility (XCF)*.

A

Application Program Interface (API)
 access 26
 description 23
Architecture 23

C

CICS
 API command 11
 API DRIVER screen 24
 API interface
 create ESS registration program 12
 event data 12
 using ESS with 11
 verify software installation 12
 configuring 23
 create ESS registration program 11
 EXEC CICS READQ TD QUEUE command 11
CICS API DRIVER screen 24
CICS API interface
 event data 11
 verify software installation 11
COMMAND entry field 24
COMMAND MSG field 24
COMMAND MSG ID field 24
COMMAND RC field 24
Commands
 EVENT SERVICES CREATE
 example 20
 format 15
 with ESS CICS API 11
 EVENT SERVICES DISPLAY
 format 22
 parameters 22
 EVENT SERVICES START
 ESS CICS API 11

Commands (*continued*)
 EVENT SERVICES START
 example 21
 format 20
 EVENT SERVICES STOP
 format 21
 parameters
 ID 22
 IMMEDIATE 22
 EXEC CICS LINK 26
COMMAREA 24, 26
Components 23
Connect:Direct
 Event Services Support
 advantages 6
 architecture 7
 concepts 5
 interfaces 6
 for CICS
 architecture 23
 components 23
Connect:Direct commands
 issuing 26
 issuing from a CICS application 23
COPY I/O start (CE)
 description 30
 type attributes 30
COPY step initiation (CI)
 description 30
 type attributes 30
COPY termination (CT)
 description 32
 event record enhancements 32

D

Data format 12, 28
Data recovery 6

Data Transmission Facility (DTF) 23

Dates

how Connect:Direct processes 17
specifying 17

DMMNHDR 12, 28

DMQ012 linking 26

DMQ247 processing 25

DMQ249 11

Driver fields, COMMAND entry 24

E

EIB 25

Entry fields

COMMAND 24
COMMAND MSG 24
COMMAND MSG ID 24
COMMAND RC 24
EVENT RESTART DATA 25
PROCESS NUMBER 25
TD EXIT COUNT 25
TS MAX LRECL 25
TS QUEUE NAME 25

ESS CICS API 11

ESS registration program 11, 12

ESS user exit 12

Event data

filtering 9, 11, 12
reading 28
what data to collect 9

Event definition 6

Event enhancements

COPY termination 32
event records 32

Event filtering 6, 9, 11, 12

Event record enhancements 32

Event record type attributes 29, 30

Event request initiation (EI)

description 30
type attributes 30

Event request stop (ET)

description 31
type attributes 31

EVENT RESTART DATA 25

EVENT SERVICES CREATE

format 15
parameters
ID 15
ORWHERE 20
WHERE 15

EVENT SERVICES DISPLAY

format 22
ID parameter 22

EVENT SERVICES START

example 21
format 20
parameters 20

EVENT SERVICES STOP

CICS API interface 11, 13
format 21
parameters 22

Event Services Support

interface 7, 11
output data format 12, 28

Event session lost (EL)

description 30
type attributes 30

Exec Interface Block 25

H

HOLD queue (QH)

description 31
type attributes 31

I

Interactive User Interface (IUI) 23

Interfaces 7, 11

N

Nonterminal tasks 23

O

Output 12, 28

P

Parameters

CASE 19
CCODE 16

Parameters (*continued*)

EXIT 19
 FNAME 19
 ID 15
 KEEPREC 21
 LASTSEQ 19, 21
 MAXQCNT 19
 ORWHERE 11, 12, 20
 PNAME 16
 PNUMBER 16
 SNODE 18
 STARTT 17, 21
 STOPT 18
 TRACE 21
 TYPE 19
 USER 18
 WHERE 11, 12, 16, 21

Process initiation (PI)

description 31
 type attributes 31

Process moved to EXEC queue (QE) 31

PROCESS NUMBER field 25

Q

Queue 28

R

Record types 29

RUN JOB start (JI)

attributes 31
 description 31

RUN TASK initiation (TI)

description 32
 type attributes 32

S

Samples

CICS API 6
 DMQ249 interface program 11
 ESS event records 10
 ESSEVX01 12
 event exit 6
 event filtering 6
 EVENT SERVICES CREATE 20
 trace of ESS records 10

Screens, Connect:Direct for CICS API DRIVER 24

Server 23

Session manager 23

System architecture

diagram 7
 Event Services Support 7

T

TD EXIT COUNT field 25

Terminal tasks 23

TIMER queue (QT)

description 31
 type attributes 31

Transient data

data flow diagram 9
 ESS data flow 8
 queue 11, 28

TS MAX LRECL 25

TS QUEUE NAME field 25

Type attributes

COPY I/O start 30
 COPY step initiation 30
 event request initiation 30
 event request stop 31
 event session lost 30
 HOLD queue 31
 Process initiation 31
 Process moved to EXEC queue 31
 RUN JOB start 31
 RUN TASK initiation 32
 TIMER queue 31
 WAIT queue 32

W

WAIT queue (QW)

description 32
 type attributes 32

