



**QVS Distributed Data Services/Controller Services
Facility for Windows**

User's Guide

Version 3

Note: Before using this information and the product it supports, be sure to read the general information under “[Notices](#)”.

First Edition (October 2004)
Updated (April 10, 2007)

This edition applies to **Version 3** of the QVS 4690 Distributed Data Services/ Controller Services Feature (**DDS/CSF**) for Windows operating systems and to all subsequent releases and modifications until otherwise indicated in new editions.

Download publications from www.qvssoftware.com. Email comments to webmaster@qvssoftware.com or address your comments to:

QVS Software, Inc.
C/o Publications
5711 Six Forks Rd. Suite 300
Raleigh, NC 27609
USA

When you send information to QVS, you grant QVS a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright QVS Software, Inc. 2004. All rights reserved.

Contents

Preface	v
Who Should Read This Manual.....	v
How This Manual Is Organized.....	v
Conventions Used in This Manual.....	vi
Related Publications.....	viii
Chapter 1. System Overview	1
Nodes	1
Node IDs	2
System ID.....	2
Logical Names and Role Names.....	2
Reserved Role Names	3
Broadcast Domains	4
Distribution Domains and Roles.....	4
File Names and Queue Names	5
Components	6
File Distribution	8
Disk I/O Prioritization.....	9
Chapter 2. Starting and Stopping DDS	10
Starting DDS.....	10
Stopping DDS	11
Stopping Abnormally.....	11
Restarting Automatically	12
Finding Out Whether DDS is Running.....	12
Starting Applications with Batch Files	13
Batch File for Start.....	13
Batch File for Stop.....	13
Batch File for Activating the Acting Primary Distributor	14
Batch File for Deactivating the Acting Primary Distributor.....	14
Example of Using Batch Files	14
Getting Help for DDS/CSF Messages	15
Chapter 3. Node Control Utility	16
Starting the Node Control Utility.....	16
Administration	17
Maintenance.....	21
Start Programs.....	22
Using the Node Control Utility from the Command Line	23
FDSADMIN.....	25
Chapter 4. Problem Analysis Utility	31
Starting the Problem Analysis Utility.....	31
DDS Configuration Files	32
DDS Dump Files.....	32
Application Function Library Configuration and Option Files	33
Windows Event Logs	33
Windows Dr. Watson Logs.....	34
Windows Memory Dump	34

System Information.....	35
User-Defined Files	35
4690 Controller Services Feature Diagnostic Files	36
Using the Problem Analysis Utility from the Command Line	37
FDSPADC.....	38
Chapter 5. Data Distribution Utility	42
Starting the Data Distribution Utility.....	42
Broadcast Domain Management.....	42
Directory Management	44
Exception Management	46
Using the Data Distribution Utility from a Command Line	48
FDSBDOMN	49
FDSGETD.....	52
FDSSETD	54
Chapter 6. Keyed File Utility	57
Starting the Keyed File Utility.....	57
Create	57
Report.....	61
Verify	63
Using the Keyed File Utility From the Command Line	64
FDSKFU	65
Hashing Algorithms	68
Polynomial Hashing Algorithm	69
Record Chaining	72
Chapter 7. DDS/CSF Provides MCF Function	74
Supported Functions and Capabilities	74
CSF Functions.....	74
DDS/CSF Supports MCF Functions.....	75
4690 Functions Not Supported or Changed in DDS/CSF	76
CSF Functions in Detail.....	79
New or Changed DDS/CSF Functions for Version 3	88
Maintaining Your 4690 OS Software	89
Problem Determination for CSF and MCF.....	90
CSF Initialization Messages	90
Helpful Tips and Limitations	90
Appendix A. Using the Keyboard Keys with DDS Utilities	93
Selecting a Menu Bar Option.....	93
Function Keys and Keyboard Keys	93
Function Keys	94
Keyboard Keys.....	94
Appendix B. Hard Disk Failure Recovery	96
Appendix C. Gathering Problem Determination Data.....	97
Notices	98
Trademarks	99
Glossary	100

Preface

This manual explains how to use the utilities and features provided with the QVS Distributed Data Services/Controller Services Feature.

Who Should Read This Manual

This manual is primarily for retail systems end users, such as store managers and information systems personnel, who are using the DDS/CSF utilities and features on the Microsoft® Windows NT®, Windows 2000®, Windows XP®, and Windows Server 2003® operating systems. This manual assumes that the reader is familiar with Windows operating systems.

How This Manual Is Organized

This manual is separated into seven chapters and four appendixes:

- [Chapter 1. System Overview](#) provides a general introduction to DDS/CSF, as well as descriptions of some concepts and terms common to all components in the DDS/CSF system.
- [Chapter 2. Starting and Stopping DDS](#) provides instructions for starting and stopping DDS, finding out whether DDS is running, and starting applications with batch files.
- [Chapter 3. Node Control Utility](#) provides instructions for starting and using the Node Control Utility, which enables you to administer the workstations associated with your DDS system.
- [Chapter 4. Problem Analysis Utility](#) provides instructions for starting and using the Problem Analysis Utility, which enables you to capture problem analysis information.
- [Chapter 5. Data Distribution Utility](#) provides instructions for starting and using the Data Distribution Utility, which enables you to manage the distribution of data in your DDS system.
- [Chapter 6. Keyed File Utility](#) provides instructions for starting and using the Keyed File Utility, which enables you to create and maintain keyed files.
- [Chapter 7. CSF and MCF](#) provides information about the 4690 Controller Services Feature and 4690 Multiple Controller Feature.
- [Appendix A. Using the Keyboard Keys with DDS Utilities](#) explains how to perform tasks within the DDS utilities using a keyboard.
- [Appendix B. Hard Disk Failure Recovery](#) provides procedures for recovering from a hard disk failure based on the role assumed by the workstation within DDS.
- [Appendix C. Gathering Problem Determination Data](#) provides steps for collecting data to aid in solving system problems.
- [Notices](#) lists legal notices required for this product.

Conventions Used in This Manual

The following conventions for terminology and syntax diagrams are used in this manual.

Terminology

For a description of the basic terms used in this book, see “[Glossary](#)” on page 101. In addition, these terms have special significance in this book:

- Command line refers to the DOS command line.
- Point to DDS refers to the folder where DDS is installed. If you choose the default installation, you will select the folder called DDS. Otherwise, point to the folder where DDS is installed.

Note: You can access the functions provided with DDS from a command line.

When using the commands, specifying the drive and directory before the command is optional if the directory where the DDS executable files are installed is in the PATH environment variable of the workstation’s system configuration, or if the command is entered from the directory where the DDS executable files are installed.

Syntax Conventions

The syntax for DDS command line commands is shown using graphic notation consisting of a statement that is tailored to the parameter requirements of each command. To read the diagrams, follow the main path line, and move from left to right and from top to bottom.

Syntax diagrams use symbols to identify the sequence of information:

- A command statement begins with:

▶▶—

and ends with:

—▶▶

- A command statement longer than one line continues to a second line with:

→

where it resumes with:

▶—

Required Parameters: A parameter that you must include is displayed on the main path line:

▶▶— COMMAND ---- variable —▶▶

If a command statement has two or more required parameters, they are shown consecutively on the main path line. A choice of required parameters is shown with one choice on the main path line and the other choices on branch lines below the first choice:

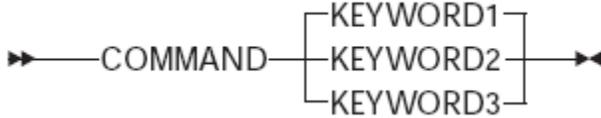
▶▶— COMMAND — KEYWORD1 — KEYWORD2 —

variable
(*)

 —▶▶

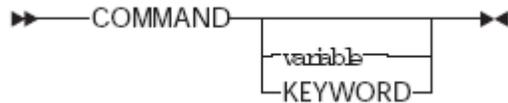
Type this command in one of two ways:
 COMMAND KEYWORD1 KEYWORD2 (variable)
 COMMAND KEYWORD1 KEYWORD2 (*)

Default Parameters: A default parameter is shown on a branch line above the main path line:



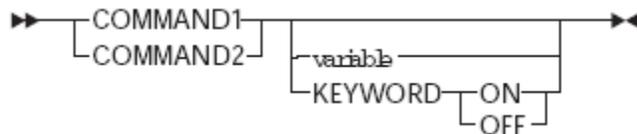
KEYWORD1 is the default.

Optional Parameters: Parameters that you can include are shown on branch lines below the main path line:



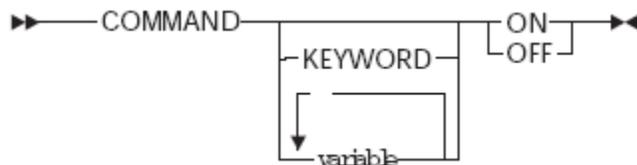
Type this command in one of the following ways:
 COMMAND
 COMMAND variable
 COMMAND KEYWORD

Branch lines can include branch lines of their own:



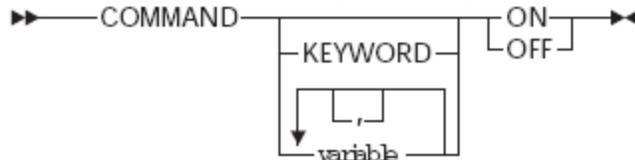
If you include the KEYWORD parameter in this statement, you must also include ON or OFF.

Repeating Parameters: An arrow on a line above a parameter means that you can repeat the parameter, or enter more than one of the listed parameters:



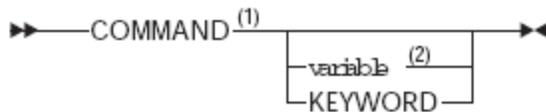
The arrow above variable means that you can include one or more values when you type COMMAND. The diagram indicates that a blank space is required between each variable value.

For commands that have optional separators between repeated values of a *variable*:



The arrow above *variable* means that you can include one or more values when you type COMMAND. This diagram indicates that a comma can optionally be placed between each variable value.

If a syntax diagram contains notes, the note numbers correspond to numbered elements shown in the diagram within parentheses:



where:

1. This is a syntax note that refers to the keyword COMMAND.
2. This syntax note describes variable.

Related Publications

In addition to this manual, the following is a list of related publications. The QVS manuals are available on <http://www.qvssoftware.com>. For more information about ordering an IBM publications, contact your IBM authorized dealer or marketing representative.

- **QVS Distributed Data Services/Controller Services Feature Version 3 Publications**

- *QVS Distributed Data Services/Controller Services Feature for Windows Installation and Configuration Guide*. This manual is available on <http://www.qvssoftware.com>.
- *QVS Distributed Data Services/Controller Services Feature for Windows Programming Guide*. This manual is available on <http://www.qvssoftware.com>.

- **IBM VisualAge® C++ for Windows Library**

- *IBM VisualAge C++ for Windows, Version 3.5 — Standard Manuals* A collection of manuals that includes: User's Guide, Programming Guide, Open Class Library User's Guide, and Visual Builder User's Guide

- *IBM VisualAge C++ for Windows, Version 3.5 — Reference Manuals A* collection of manuals that includes: Visual Builder Parts Reference, Building VisualAge C++ for Fun and Profit, Language Reference, C Library Reference, Open Class Library Reference Volume I, II, III, and IV, SOM Programming Guide, and SOM Programming Reference

- **C-Related Publications**

- *Portability Guide for IBM C*

- **4690 OS-Related Publications**

- *4690 OS Version 3: Planning, Installation, and Configuration Guide*

Chapter 1. System Overview

The QVS Distributed Data Services/Controller Services Feature (DDS/CSF) is a distributed software platform designed as a base for the development of distributed applications for the store environment.

The main functions are:

- Interprocess communications (IPC) with local and remote transparency
- Data access with local and remote transparency
- Data distribution for redundancy, performance, and availability
- Directory services
- Installation, configuration, and administration

The following sections provide an overview of the concepts that are common to all components of DDS/CSF:

- [“Nodes”](#) below
- [“Logical Names and Role Names”](#) on page 2
- [“Broadcast Domains”](#) on page 4
- [“Distribution Domains and Roles”](#) on page 4
- [“File Names and Queue Names”](#) on page 5
- [“Components”](#) on page 6
- [“File Distribution”](#) on page 8
- [“Disk I/O Prioritization”](#) on page 9

Nodes

A node is a LAN-attached machine that is running DDS.

Nodes can be connected via LANs to form a system. A system is the group of nodes for which files are managed. A node can be connected to a subset of nodes via one LAN and to other subsets of nodes via different LANs. Data distribution, remote file access, and interprocess communication are supported only among nodes that are connected by a LAN. DDS does not provide a bridge or router function.

For all system topologies with more than one node, one node must be installed and configured as the configured primary distributor. You can optionally install and configure another node as the configured backup distributor. The backup distributor is used to add redundancy to the critical store data. Each node must be connected to the primary distributor and backup distributor by a LAN. See [“Distribution Domains and Roles”](#) on page 4 for an explanation of primary distributor, backup distributor, and subordinate.

Node IDs

Node IDs are specified for each workstation (node) during the installation of DDS, and assumed each time DDS initializes. Each node has a single node ID, regardless of how many LANs are used to connect it to other nodes.

The rules for node IDs are:

- Each workstation within a system must have a different node ID. Duplicate node IDs are not confirmed during installation, but they are detected when DDS initializes. If duplicate node IDs are detected, you must reinstall DDS to correct the problem.
- No two workstations on a LAN can have the same node ID even if they are members of different DDS systems.
- Node IDs can be from 1 to 8 characters (blanks are not allowed). Greater than and less than characters (< and >), question marks (?), asterisks (*), and colons (:) are not valid characters for node IDs.
- Node IDs are case-sensitive.
- Do not define node IDs whose first three characters are FDS. This prefix is reserved.
- Do not use node IDs whose first character is a / (forward slash).
- After installation, node IDs can be changed only by reinstalling DDS.

System ID

Each system has a 4-byte system ID. The system ID is specified at each node during the installation of DDS, and defaults to 0000. The Name Services component uses the system ID as a qualifier when locating the node that has assumed a particular role. See “[Logical Names and Role Names](#)” for a description of roles. This method ensures, for example, that each node finds the acting primary distributor for its system, and that files are distributed only within a system. The system ID must be set to a unique value for each system in environments where multiple systems are interconnected via bridges. A system ID cannot be changed without reinstalling DDS.

If you assign a different system ID to each system, the systems can be connected using a LAN or a gateway without causing messages intended for one system to be sent to another system.

If you assign the same system ID to each system, the LANs can be connected using a gateway; both LANs are considered the same system. If this type of system is used and you are using the Data Distribution component, there should only be one configured primary distributor and one configured backup distributor for the entire system, even though they might be on different LANs.

Logical Names and Role Names

DDS provides a name-resolution capability, allowing applications to use logical names instead of hard-coded file names, interprocess communication (IPC) queue names, and node IDs. These logical names are dynamically resolved when the application runs.

Some names are fairly static over time. An example is the name of a configuration file. Although this name is not likely to change very often, if ever, it is still desirable to avoid using the name in an application program. The use of a logical name allows the file name to be changed without having to rebuild the application. A logical name has the following format:

`<name>`

where *name* is 1 to 260 characters and the less than and greater than characters (< and >) are required delimiters.

Other names are more dynamic, such as the node ID of the primary distributor. This changes whenever the backup distributor takes over for the primary distributor. In this case a role (the primary distributor) is assumed by a particular node. A logical name can be used to identify this role and is referred to as the role name. A role name has the following format:

`<name::>`

where *name* is 1 to 8 characters, the less than and greater than characters (< and >) are required delimiters, and double colons (::) indicate that this is a role name.

The use of a role name makes it easy for an application to open a file or IPC queue on a node that provides a particular service when the service may move from node to node as conditions change. The primary distributor is an example of a service provided by DDS. Applications can be written that provide other services and role names can be defined for them.

Reserved Role Names

The following role names are reserved by DDS and are used to identify the primary distributor and backup distributor nodes. These names are dynamically maintained by DDS and cannot be modified by the user.

Role Name: — Reserved for:

:

FDSFDXCP:

Configured primary distributor

FDSFDXCB:

Configured backup distributor

FDSFDXAP:

Acting primary distributor

FDSFDXAB:

Acting backup distributor

Broadcast Domains

A subset of nodes within a system can be grouped into a broadcast domain. A broadcast domain has a name, the broadcast domain name, which has these rules:

- Broadcast domain names can be from 1 to 8 characters (blanks are not allowed). Greater than and less than characters (< and >), question marks (?), asterisks (*), and colons (:) are not valid characters.
- Broadcast domain names are case-sensitive.
- Do not use broadcast domain names whose first three characters are FDS. These names are reserved.
- DDS currently supports a maximum of one broadcast domain.

Broadcast domains are useful for maximizing performance and minimizing resource utilization when distributing messages or files to a large number of nodes. Within a broadcast domain, DDS exploits the LAN hardware's ability to broadcast a datagram to all nodes. The DDS Data Distribution component supports distributing files to all nodes within a broadcast domain. The IPC component can send a message to every node within a broadcast domain.

Distribution Domains and Roles

Each node can assume a distribution role. There are three possible distribution roles:

Primary distributor

The primary distributor controls the primary copy of all distributed files. Only the primary copy of a file can be modified directly by an application.

Backup distributor

The backup distributor controls the backup copy of a file and can take over for the primary distributor if the primary distributor fails or is deactivated.

Subordinate

All other nodes configured with the Data Distribution component are considered subordinates and manage image copies of distributed files.

A node can be configured with a distribution role, but under certain circumstances can assume another role. In that instance, the node is said to be *acting the role*. Specifically, when a node that is configured as the backup distributor assumes the role of the primary distributor that has failed, the backup distributor is said to be the *acting primary distributor*.

A group of nodes form a distribution domain. There are two types of distribution domains:

Mirrored domain

A mirrored domain is defined to be the primary distributor and backup distributor. There is only one mirrored domain, so it is not named.

Broadcast domain

A broadcast domain can include zero (0) or more nodes. Files that are distributed to a broadcast domain are distributed to each node in the domain, as well as to the acting backup distributor.

A given file can be distributed to only one domain.

By default, all files on a node are local files. A local file is a file that is not distributed. That is, it is not a primary copy, backup copy, or image copy. A file or subdirectory on the acting primary distributor can be made distributed using either the Data Distribution Utility or an application that calls the `FdsSetDistribution()` API. (Refer to QVS Distributed Data Services/Controller Services Feature for Windows Programming Guide for more information about the `FdsSetDistribution()` API.)

File Names and Queue Names

For a specific node, you can identify each file using the operating system path where the file is located and the file name. The operating system path and file name are called the file specification. Similarly, you can identify a queue using the queue name.

However, with DDS you can access files and queues on any node. Therefore, you must use a retail path specification to identify the file or queue. A retail path specification contains a node specification or broadcast domain specification as a prefix to the file specification or queue name.

Note: If you do not include a node or broadcast domain specification in the retail path specification, DDS assumes that the file or queue resides locally.

A node specification can be in one of these forms:

- A node name followed by two colons, which specifies the ID of the node directly.
For example:
NodeID::
- A role name followed by two colons and delimited with the greater than and less than signs. The role name will resolve to the ID of the node that is acting in the role.
For example:
<RoleName::>

A broadcast domain specification is valid only to identify a queue. It must be a string, containing no blank characters, that includes a broadcast domain name followed by two colons. For example:

B_DOMAIN::

Assume that you have a file called MYFILE.DAT located in the subdirectory D:\FILES on the primary workstation with a node ID of Node1. The following retail path specifications are valid:

```
D:\FILES\MYFILE.DAT
Node1::D:\FILES\MYFILE.DAT
<FDSFDXAP::>D:\FILES\MYFILE.DAT
```

You can also use logical names for any part (or all) of a retail path specification. See “[Logical Names and Role Names](#)” on page 2 for more information about logical names.

Note: If you use a logical name for a file name or queue name in a retail path specification that contains a remote node ID or role name, the logical name is resolved on the remote node.

Components

The components for DDS/CSF are:

Name Services

Provides a name-resolution capability. This capability allows applications to use logical names, or aliases, instead of hard-coded file names, IPC queue names, and node IDs. These logical names are dynamically resolved at run time.

Refer to the *QVS Distributed Data Services/Controller Services Feature for Windows Programming Guide* for more information about the Name Services component.

File Services

Provides access both local and remote files. File Services can be optionally configured on zero or more nodes to share files with other nodes. The configured primary distributor and configured backup distributor are configured to share retail files by default.

Three types of files are supported:

- Keyed files
- Sequential files
- Binary files

Refer to the *QVS Distributed Data Services/Controller Services Feature for Windows Programming Guide* for more information about the File Services component.

File System Interface

Provides support for distributing native operating system files, referred to as byte stream files.

This component also provides a disk I/O prioritization mechanism that overrides the standard operating system prioritization scheme. DDS prioritizes disk I/O based on thread priority, allowing high priority requests, such as price lookups, to be processed ahead of lower priority requests.

Interprocess Communications

Provides a peer-to-peer messaging service that allows application programs to send and receive messages. The messaging service is provided between processes running on a single node (intranode IPC) and on different nodes (internode IPC). The internode IPC function is a configuration option. It is required on all nodes in a system unless the system consists of a single standalone node.

The following LAN media are supported:

- Ethernet
- Token ring
- IBM Wireless LAN

Note: If you install the DDS/CSF 4690 Controller Services Feature, DDS/CSF also provides store loop support.

Refer to the *QVS Distributed Data Services/Controller Services Feature for Windows Programming Guide* for more information about the Interprocess Communications component.

Data Distribution

Provides a distributed file capability that replicates data to multiple nodes, keeping each image synchronized during normal operations. It also performs reconciliation when failed nodes are returned to service. You can use these functions through a utility. Refer to the *QVS Distributed Data Services/Controller Services Feature for Windows Programming Guide* for more information about the Data Distribution component. See “[Chapter 5. Data Distribution Utility](#)” on page 42 for more information about the utility.

Node Control

Provides administrative functions such as viewing information about the nodes within the DDS system and activating or deactivating the primary distributor. These aspects of node control are used through a utility. Node control is also responsible for synchronizing the time and date of all nodes within a system.

There is also an API that generates a list of all node IDs known to the DDS system, including nodes that DDS has detected as being active on the system and user-defined nodes that are not yet active.

Refer to the *QVS Distributed Data Services/Controller Services Feature for Windows Programming Guide* for more information about using the node list API. See [Chapter 3. Node Control Utility](#) for more information about using the Node Control Utility.

Problem Determination and Analysis

The Problem Determination and Analysis component collects problem determination information. You can select the system message logs, system error logs, and system dump files you wish to work with. See “[Chapter 4. Problem Analysis Utility](#)” on page 31 for more information.

4690 Controller Services

Provides support for 4680/4690 Operating System controller applications running under Windows, and the attachment of registers running the 4690 Operating System. It also supports the attachment of registers running DOS and the 4690 Terminal Services for DOS feature. This optional feature can be installed on zero or more nodes. Refer to the *QVS Distributed Data Services/Controller Services Feature for Windows Installation and Configuration Guide* for more information about 4690 Controller Services.

4690 Multiple Controller Feature

This optional feature can be installed on zero or more nodes. It provides support for 4690 Controller Services Feature nodes to interact with other 4690 Controller Services Feature nodes. It also provides support for 4690 terminal backup in a multi-node environment. Refer to the *QVS Distributed Data Services/Controller Services Feature for Windows Installation and Configuration Guide* for more information about 4690 Multiple Controller Feature.

File Distribution

DDS enables the distribution of files to other nodes in a distribution domain. When a file operation is directed to a controlled drive, DDS determines if the file has been defined as a distributed file or is in a distributed subdirectory, and then distributes the file operation as appropriate to other nodes.

File distribution is performed with no user intervention. Any operating system command (for example, **COPY** or **ERASE**) or application program statement that results in the modification of a distributed file causes DDS to distribute the operation to other nodes.

DDS must be running to detect whether a file is distributed. Until DDS is started, it assumes that all files on a controlled drive are distributed. Therefore, any attempt to modify a file on a controlled drive when DDS is not running results in an error.

Note: If the **DDActive** configuration keyword is set to **NO**, DDS does not check to see if files on controlled drives are distributed, so this error is not returned. Refer to the Configuration Keywords chapter in the *QVS Distributed Data Services/Controller Services Feature for Windows Installation and Configuration Guide* for more information about the **DDActive** configuration keyword.

Refer to the Data Distribution chapter in the *QVS Distributed Data Services/Controller Services Feature for Windows Programming Guide* for more information about

distributing files. Refer to the Planning for DDS/CSF chapter in the *QVS Distributed Data Services/Controller Services Feature for Windows Installation and Configuration Guide* for more information about controlled drives.

Disk I/O Prioritization

Processing of certain file I/O operations is extremely time critical in the retail environment. For example, when scanning items at the point-of-sale terminal, the salesperson expects a consistent response time. If the processing load on the disk used to service price lookup requests increases, the response time experienced by salespeople scanning items at the point-of-sale terminals should remain relatively constant.

DDS uses a disk I/O prioritization scheme to assure that time-critical processes are given highest priority for disk access. This prioritization scheme overrides the prioritization scheme of the operating system, so that file access is granted based on the operating system priority of the process thread issuing the request.

To take advantage of disk I/O prioritization, an application program can increase the priority level of time-critical threads using the `SetThreadPriority()` API on Windows.

Disk I/O prioritization is most effective when all partitions on a single physical disk are controlled by DDS. This fact should be considered when configuring partitions as controlled drives. Refer to the Planning for DDS/CSF chapter in the *QVS Distributed Data Services/Controller Services Feature for Windows Installation and Configuration Guide* for more information about controlled drives.

Chapter 2. Starting and Stopping DDS

This chapter provides information about starting and stopping DDS. It also explains how to determine whether DDS is running on a particular workstation. You can find more information in these sections:

- “Starting DDS” below
- “Stopping DDS” on page 11
- “Stopping Abnormally” on page 11
- “Restarting Automatically” on page 12
- “Finding Out Whether DDS is Running” on page 12

This chapter also provides information about creating batch files for automatically starting and stopping applications that run with DDS. See “[Starting Applications with Batch Files](#)” on page 13 for more information.

Starting DDS

When DDS is installed on a local workstation, it is installed as a Windows service that is started automatically after each IPL of the operating system. If you do not want DDS to start automatically after each IPL, use the Windows Services panel to specify that it be started manually.

Note: DDS is set up to automatically log on to a system account and to interact with the desktop. If these options are not activated, then DDS will not operate correctly.

You can also start DDS using the **net start** command. To start DDS using this command, type **net start dds** at the command line.

Note: The user ID that you use to log on to the workstation must be a member of the Windows administrators group for that workstation in order to start DDS from the Windows Services panel or by using the **net start dds** command.

When DDS is started as a Windows service, it may not be readily apparent that DDS is started; the product name will not be listed in the Windows task list. DDS continues running even when there are no users logged on to the workstation.

Only one instance of DDS can run on a workstation at any given time. If DDS is started from the command line when another instance of DDS is already running, a message is displayed stating that DDS is already running.

If DDS cannot initialize successfully because of an internal error or an operating system error, the Windows operating system will display a message indicating that DDS initialization failed, and the error will be logged in the Windows event logs.

Please note that, if you are running DDS under Windows XP with Service Pack 2 and you have activated the Windows Firewall, you need to do one of two things in order to run successfully with DDS.

1. Disable the Windows Firewall or
2. Enable the Windows Firewall, but configure an exclusion for the following programs:
 - c:\fds\bin\fdsiprt.exe
 - c:\fds\bin\fdst286.exe
 - c:\fds\bin\dhcpd.exe

Note: The path of the above exclusion could be different if you have installed DDS in another directory. If so, you should adjust the path accordingly.

There are some circumstances when DDS can restart itself. See “[Restarting Automatically](#)” on page 12 for more information.

Stopping DDS

You can stop DDS using either the Windows Services panel or the Windows **net stop command**. To stop DDS using the **net stop** command, type **net stop dds** at the command line.

Note: The user ID that you use to log on to the workstation must be a member of the Windows administrators group for that workstation in order to stop DDS from the Windows Services panel or by using the **net stop dds** command.

You can also stop DDS using the Node Control user interface. See “[Stop DDS](#)” on page 17 for more information.

DDS also stops automatically under some circumstances. See “[Stopping Abnormally](#)” for more information.

When a normal stop command is received, DDS processes all outstanding requests before it stops, but it will not accept any new user requests or API calls. If DDS does not stop successfully because of an internal error or an operating system error, the error will be logged in the Windows event logs.

Stopping Abnormally

An abnormal stop occurs when DDS encounters an error condition from which it cannot recover. DDS/CSF stops processing any outstanding requests, completes only the functions necessary for it to restart normally, and stops.

Restarting Automatically

DDS restarts automatically after an abnormal stop unless user intervention is required to correct the condition that caused the abnormal stop. When an automatic restart is not possible, a message is logged in the Windows event logs that notifies the user of the condition that caused the failure.

Finding Out Whether DDS is Running

Because DDS runs as a Windows service, it does not appear in the task list after it has been started. Therefore, it is not always obvious whether DDS is running unless you refer to the Windows Services panel. Furthermore, the Services panel does not contain information about what state the product is in.

The command **FDSQUERY** verifies whether DDS has been started on the local workstation. When you type this command at the command line, a message is displayed that tells you whether DDS is running or if it is in the process of starting or stopping.

To find out whether DDS is running on the local workstation, from the command line, type **FDSQUERY** and press **Enter**.

Note: If the directory where DDS executables are located is in the PATH= statement in the workstation's system configuration, DDS can be queried from any directory on that workstation. Otherwise, you must enter the command from the directory where DDS is installed.

The **FDSQUERY** command sets a return code that can be used by applications that call this command to determine what state DDS is in. This return code is not included in the messages displayed by **FDSQUERY**.

The following return codes can be returned from the **FDSQUERY** command when it is called by an application. Also shown are the associated messages that are displayed when you enter the **FDSQUERY** command from the command line:

- | | |
|-----------|-----------------------------------------------------------|
| 1 | DDS not running |
| 2 | DDS stopping |
| 3 | Command canceled by user |
| 4 | Internal error |
| 5 | Operating system error |
| 13 | DDS has been started successfully and is now initializing |
| 20 | DDS running |
| 99 | Command canceled by user |

If the **FDSQUERY** command is not successful because of an internal error or an operating system error, the error will be logged in the Windows event logs.

Starting Applications with Batch Files

Under certain circumstances, DDS starts a specific batch file so that applications required to run under these circumstances can be started automatically. Default copies of the batch files are installed by DDS into the subdirectory containing the product executables; however, the batch files do not need to exist to run DDS.

DDS starts a separate batch file on the local workstation under each of the following conditions:

- DDS starts on the local workstation.
- DDS stops on the local workstation.
- The local workstation is activated as the acting primary distributor.
- The local workstation is deactivated so that it is no longer the acting primary distributor.

Note: DDS does not verify whether an application specified in a batch file is already running, and therefore cannot prevent multiple instances of applications from being started by these batch files.

See the appropriate Windows documentation for information on writing batch files.

Batch File for Start

After DDS starts successfully, it asynchronously starts the batch file FDSBEGIN.BAT. If you choose to use batch files, add instructions to this file that you want to execute when DDS starts, such as starting any applications that require DDS to be running.

Batch File for Stop

Whenever DDS begins a normal or an abnormal stop, it starts the batch file FDSSEND.BAT before stopping the DDS components. If you choose to use batch files, add instructions to this file that you want to execute when DDS stops, such as running any commands to stop applications that depend on DDS.

When DDS stops normally, it waits until the batch file finishes running before continuing the stop process in order to allow applications with dependencies on DDS to be able to finish running before DDS stops.

When DDS stops abnormally, it does not wait for the batch file to finish running before it stops, because it might not be able to process any more requests or API functions without errors.

Note: This batch file is not started when DDS stops as a result of the Windows operating system shutting down. You should stop any applications requiring DDS before shutting down the operating system.

Batch File for Activating the Acting Primary Distributor

DDS asynchronously starts the batch file FDSAP.BAT after a workstation is activated as the acting primary distributor. This includes the instances when DDS starts on the acting primary distributor. If you choose to use batch files, add instructions to this file that are required to be executed after the acting primary distributor is activated.

Batch File for Deactivating the Acting Primary Distributor

DDS starts the batch file FSDSP.BAT whenever the acting primary distributor is deactivated. This includes the instances when DDS stops on the acting primary distributor. If you choose to use batch files, add instructions to this file that are required to be executed before the acting primary distributor is deactivated.

DDS waits until this batch file finishes running before deactivating the acting primary distributor in order to allow applications with dependencies on the acting primary distributor to finish running.

Note: This batch file is not started when a workstation relinquishes the acting primary distributor role as a result of the Windows operating system shutting down. You should stop any applications requiring DDS before shutting down the operating system.

Example of Using Batch Files

The following example is for using the FDSBEGIN.BAT batch file that is provided with DDS:

```
myprog1 myparam1
start cmd.exe /c myprog2 myparam1 myparam2
call mybat1 myparam1
start cmd.exe /k mybat2 myparam1 myparam2 myparam3
```

myprog1 myparam1 runs a program called MYPROG1.EXE with a parameter myparam1. Because the **start** command was not used in this command, the program MYPROG1.EXE will run in the same window as FDSBEGIN.BAT. As a result, the next command in the FDSBEGIN.BAT file will not be processed until MYPROG1.EXE stops running. This method for starting programs should be used when a program must complete processing before a subsequent program can be started.

start cmd.exe /c myprog2 myparam1 myparam2 starts the program MYPROG2.EXE with parameters myparam1 and myparam2 in a separate window. The MYPROG2.EXE program will be started in a separate window and will also be listed in the Windows task list. The /c parameter instructs Windows to close the window in which MYPROG2.EXE is running after MYPROG2.EXE stops. Because the Windows **start** command is used to start

MYPROG2.EXE, FDSBEGIN.BAT will not wait for MYPROG2.EXE to finish before processing the next command in the file.

call mybat1 myparam1 runs the batch file MYBAT1.BAT with a parameter myparam1. The **call** command is used to run batch files from within other batch files, such as FDSBEGIN.BAT. Because the **call** command is used, FDSBEGIN.BAT will wait for MYBAT1.BAT to stop running before processing the next command.

start cmd.exe /k mybat2 myparam1 myparam2 myparam3 starts the batch file MYBAT2.BAT with parameters myparam1, myparam2, and myparam3 in a separate window. Like the MYPROG2.EXE program in the third command, MYBAT2.BAT will be started in a window and will also be listed in the task list. Unlike that command, however, the /k parameter instructs Windows to keep the window open after MYBAT2.BAT stops. Because the **start** command is used to start MYBAT2.BAT, FDSBEGIN.BAT will not wait for MYBAT2.BAT to finish before exiting the program.

Note: Always use the commands **start** or **call** when starting batch files, such as MYBAT1.BAT and MYBAT2.BAT, from within other batch files like FDSBEGIN.BAT. If you start a batch file by itself (without using **start** or **call**), control is passed to the called batch file but never returned.

For more information about the **start** and **call** commands and parameters, refer to the appropriate Windows documentation. You can also type help start or help call from the command line to view information about the **start** and **call** commands online.

Getting Help for DDS/CSF Messages

DDS/CSF logs messages to alert you to the status of a request or to an error that has occurred. Any errors that affect the operation of DDS/CSF are logged in the Windows event logs.

For all DDS/CSF messages, you can use **FDSHELP** to obtain an explanation and recommended actions. To use **FDSHELP**, type **FDSHELP** followed by the message number and press Enter (for example **FDSHELP fds0001**). The message is displayed along with any actions that should be taken by the user or store support representative as a result of the error.

Note: All DDS messages begin with the characters fds.

Chapter 3. Node Control Utility

The Node Control Utility enables you to administer the workstations associated with your DDS/CSF system. All of the Node Control Utility functions can be initiated from a panel-driven graphical user interface or from a command line. See “[Starting the Node Control Utility](#)” for more information about using the graphical user interface. See “[Using the Node Control Utility from the Command Line](#)” on page 21 for more information about taking action from the command line.

Note: DDS must be running on any remote workstation that you specify in a Node Control function for that function to be successful.

Starting the Node Control Utility

DDS must already be running on the workstation before you can start the Node Control Utility.

Two modes are provided for accessing the Node Control Utility:

- **Administrator mode**

This mode provides an interface for entering Node Control administrative commands in addition to viewing the Node Control Utility report. To start the Node Control Utility in Administrator mode, click **Start** on the taskbar, point to **Programs**, point to **DDS**, and select **Node Control**. You can also type **FDSNCU /ADMIN** from the command line and press **Enter**.

The main panel of the Node Control Utility in Administrator Mode is displayed with four menu options:

- Administration, which allows you to control DDS system functions.
- Maintenance, which allows you to add or delete node IDs from the node status list.
- Start Programs, which allows you to start programs on remote nodes.
- Help, which provides help for the panels.

There are also two status fields:

- Command Request, which displays the status of actions you have initiated.
- Command Status Messages, which displays any messages that result from the actions you have initiated.

- **Report mode**

This mode provides access to the Node Control Utility report only. To start the Node Control Utility in Report mode, type **FDSNCU** from the command line and press **Enter**. The Node Control Reports panel is displayed, which contains the following information:

- Time and date on the local workstation
- Node ID of the local workstation

- Node roles of the local workstation
- A list of active workstations
- A list of inactive workstations

See “[Node Control Report](#)” on page 18 for more information about using the Node Control Utility in Report mode.

The following return codes can be returned from the **FDSNCU** command when it is called by an application. Also shown are the associated messages that are displayed when you enter the **FDSNCU** command from the command line:

- | | |
|-----------|--------------------------|
| 0 | Operation was successful |
| 1 | DDS is not running |
| 3 | Command canceled by user |
| 4 | Internal error |
| 5 | Operating system error |
| 99 | Command canceled by user |

Administration

You can use the Administration functions to control DDS system functions on workstations that are running DDS. To use the Administration functions, select **Administration** from the Node Control main panel. The following options are displayed:

- Stop DDS
- Set Date and Time
- Node Control Report
- Deactivate Acting Primary
- Distributor Activate Primary
- Distributor Dump DDS

Stop DDS

You can stop DDS on a selected workstation or on all workstations. The stop process on remote workstations is the same as if a user at the remote workstation had initiated the stop. To stop DDS on one or more workstations:

1. Select **Stop DDS...** from the Administration options pull-down.
2. Select one or more node IDs of the workstations on which you want to stop DDS. You can select all node IDs in the list by selecting **Select All**, or deselect all node IDs in the list by selecting **Deselect All**.
3. Select **Accept**.

See “[Stopping DDS](#)” on page 11 for additional information about stopping DDS.

Set Date and Time

You can set the date and time for the entire DDS system. To set the date and time for the DDS system from a workstation:

1. Select **Set Date and Time...** from the Administration options pull-down.
2. Change the current values in the Month, Day, Year, Hour, and Minute fields to the desired values.
3. Specify whether the time is a.m. (between midnight and noon) by selecting the **am** check box. Do not select the box if you want to specify a p.m. time.
4. Select **Accept**.

If you set the date and time on a local workstation that is not communicating with the acting primary distributor, the date and time are set for the local workstation only. When the workstation reestablishes communication with the acting primary distributor, the date and time on the local workstation will be synchronized with the date and time on the acting primary distributor.

- DDS synchronizes the time and date for all nodes that are communicating with the acting primary distributor:
- Each time a workstation establishes a connection to the acting primary distributor.
- Whenever the time or date is changed using the Node Control Utility.
- Every day at 3:00 a.m.

Note: If you use the Windows system commands to set the time and date, the time will not be synchronized on all workstations.

Node Control Report

The Node Control Report displays information about your DDS system, such as the node ID, role names, and the system time and date for a workstation, as well as the connection status for all workstations in the DDS system.

Note: The Acting Primary Distributor maintains the list of workstations in your DDS system and the connection status for those workstations. Therefore, the Node Control Report can be viewed only if the acting primary distributor is online and active. Otherwise, a message is returned stating that the status of the workstation is not available.

To view the Node Control Report:

1. Select **Node Control Report** from the Administrative options pull-down.
The Node Control Report displays the following information:
 - Time and date on the local workstation
 - Node ID of the local workstation
 - Node roles of the local workstation

- Node status list, which lists all active and inactive workstations in your DDS system

Note: The status information is current at the time that it is displayed, and is updated each time the screen is refreshed.

2. Select Refresh to see any new information.
3. Select Cancel when you have finished viewing the Node Control Report.

See “[Add Node IDs to the Node Status List](#)” on page 21 and “[Delete Node IDs from the Node Status List](#)” on page 22 for more information about the node status list.

Activate Primary Distributor and Deactivate Acting Primary Distributor

The primary task of the configured backup distributor is to assume control when the configured primary distributor becomes disabled or is deactivated. The first time the system is started, the configured primary distributor assumes the acting primary distributor role. It remains the acting primary distributor until it becomes disabled or is deactivated.

The default behavior of DDS results in the acting backup distributor not assuming the acting primary distributor role automatically. In this case, two steps are required before the acting backup distributor can be activated as the primary distributor.

1. Deactivating the acting primary distributor
2. Activating the configured backup distributor as the acting primary distributor

You must deactivate the current acting primary distributor before you can activate a new primary distributor, unless the current acting primary distributor is not running or is not communicating with other workstations. Refer to *Distributed Data Services for Windows: User's Guide* for more information about using the Node Control Utility to perform these deactivation and activation steps.

Note: If the configured primary distributor will be disabled for only a short period of time, you may not need to activate the configured backup distributor as the acting primary distributor. This might be the case if the applications running on other nodes use image copies of input files and asynchronously write output data to the acting primary distributor.

You must deactivate the current acting primary distributor before you can activate a new primary distributor, unless the current acting primary distributor is not running or is not communicating with other workstations.

When the configured primary distributor resumes normal operation (it is powered on and connected to the LAN), it does not automatically resume its role as the acting primary distributor. If the configured backup distributor was activated by the operator as the acting

primary distributor, the configured primary distributor assumes the acting backup distributor role.

To return the system to its normal state, you must first deactivate the acting primary distributor (the configured backup distributor) and then re-activate the configured primary distributor as the acting primary distributor.

More complex scenarios are possible. For example, the acting primary distributor could fail, the operator could activate the acting backup distributor as the acting primary distributor, and then the new acting primary distributor could fail. At this point, there are several possibilities:

- The new acting primary distributor could resume normal operation. In this case, no data should be lost.
- The original acting primary distributor could resume normal operation before the new acting primary distributor. In this case, the original acting primary distributor resumes its role as the acting primary distributor. Data will be lost that was collected (updated) on the new acting primary distributor after the original acting primary distributor failed.
- Both the original acting primary distributor and the new acting primary distributor could resume normal operation simultaneously. In this case, the new acting primary distributor detects that it assumed the role of the acting primary distributor more recently than the original acting primary distributor, and again assumes the acting primary distributor role. No data should be lost in this case.

Deactivate Acting Primary Distributor: To deactivate the acting primary distributor:

1. Select **Deactivate Acting Primary Distributor** from the Administration options pull-down.
2. Select **Yes** to deactivate the acting primary distributor or **No** to cancel the request.

Activate Primary Distributor: To activate the acting primary distributor:

1. Select **Activate Primary Distributor** from the Administration options pull-down.
2. Type the node ID of the workstation that will become the acting primary distributor.
3. Select **Force if the backup not reconciled** to force the workstation to activate as the acting primary distributor. A check in the box means that the workstation will activate as the acting primary distributor even if its files are not reconciled with the old acting primary distributor.

Attention: Forcing the workstation to activate as the acting primary distributor when its files are not reconciled with the old acting primary distributor could result in data being lost.

4. Select **Accept**.

Dump DDS

This function performs a system dump of DDS on the specified workstations at the time the command is received. It is intended for use when DDS is not operating correctly.

To force a system dump of DDS:

1. Select **Dump Distributed Data Services...** from the Administration options pull-down.
2. Select one or more node IDs of the workstations on which you want to perform a system dump. You can select all node IDs in the list by selecting **Select All**, or deselect all node IDs in the list by selecting **Deselect All**.
3. Select **Accept**.

The dump process on remote workstations is the same as if a user at the remote workstation had initiated the dump. When the dump is complete, use the Problem Analysis Utility to collect the dump. Contact your QVS representative for more information about using the dump information.

Maintenance

You can use the Maintenance functions to add Node IDs to and delete Node IDs from the node status list. To use the Maintenance functions, select **Maintenance** from the main panel. The following options are displayed:

- Add Node IDs to the Node Status List
- Delete Node IDs from the Node Status List

Add Node IDs to the Node Status List

DDS maintains a list of node IDs and their connection status with the acting primary distributor for the DDS system. When a workstation establishes communication with an acting primary distributor for the first time, DDS automatically adds its node ID to this list and records the workstation's status as "Active."

The node status list is updated by DDS on the acting primary distributor each time DDS is initialized on a workstation. After a workstation has been added to the list, either by DDS or by you, the workstation and its corresponding status information are maintained in the list until you delete them.

The status for a workstation is changed to "Inactive" if that workstation stops communicating with the acting primary distributor (for example, if the workstation shuts down, if the connection between the workstation and the acting primary distributor is down, or if DDS is not active on the workstation).

The Add Node IDs to the Node Status List function allows you to add node IDs of inactive nodes to the node status list. These node IDs can be used as place holders for future workstations, or for workstations that are not currently configured to communicate with the acting primary distributor. The status for any added workstation is “Inactive” until the workstation actually establishes communication with the acting primary distributor. At that time, it is set to “Active.”

To add node IDs to the node status list:

1. Select Add Node IDs to the Node Status List... from the Maintenance options pull-down.
2. Type one or more node IDs to indicate which workstations you want to add. Use a space as a separator if you are typing more than one node ID.
Note: Node IDs are case-sensitive.
3. Select **Accept**.

Delete Node IDs from the Node Status List

The Delete Node IDs from the Node Status List function allows you to delete node IDs that are no longer valid from the node status list. Only node IDs for workstations with a connection status of “Inactive” can be deleted from the node status list.

A workstation with a deleted node ID will automatically be added again into the node status list if communication with the acting primary distributor is reestablished.

A deleted node ID can also be added again into the node status list by using the Add Node IDs to the Node Status List function. After the node ID is added back into the node status list, status information about the workstation associated with that node ID will be available in the Node Control Report.

To delete node IDs from the node status list:

1. Select **Delete Node IDs from the Node Status List...** from the Maintenance options pull-down.
2. Select one or more node IDs that you want to delete. You can select all node IDs in the list by selecting **Select All** or deselect all the node IDs in the list by selecting **Deselect All**.
3. Select **Accept**.

Start Programs

This function starts a program on the specified workstations. Both programs and batch files can be started using this function.

Note: You cannot specify DDS as the program to be started, because it must already be running on the specified workstations to process the request.

To start a program on a specified workstation:

1. Select **Start Programs** from the Node Control main panel.
2. Select **Start a Program on a Remote Workstation...** from the Start Programs options pull-down.
3. Type the name of the program to be started.
Program names require the extension to be specified; for example, you would specify PROGRAM.BAT or PROGRAM.EXE. The name must be a valid program name that exists on a disk drive accessible from the remote workstation, and the program must be located in a subdirectory that is in the PATH= statement of the remote workstation's system configuration.
Note: To start a program that is a batch file:
 - a. Specify the program name as CMD.EXE.
 - b. Specify either /c or /k as a parameter. Specify /c to close the Windows window after the batch file stops. Specify /k to keep the window open after the batch file stops (provided the batch file does not end with an **EXIT** command).
 - c. Specify the full name of the batch file as a parameter.
 - d. Specify any additional parameters.
4. Type any program parameters. Use spaces as separators (for example, **MYBAT.BAT parameter1 parameter2**).
5. Select one or more workstations from the provided list of node IDs. You can select all node IDs in the list by selecting **Select All** or deselect all node IDs in the list by selecting **Deselect All**.
6. Select **Accept**.

You will receive notification if the start program command was successful on the specified workstations, but no further information will be provided about that program.

Using the Node Control Utility from the Command Line

All of the functions for the Node Control Utility can also be initiated from the command line. Functions that can be performed include:

- Activate or deactivate the acting primary distributor.
- Set the time and date for all workstations.
- Add a workstation to a list of system nodes.
- Delete a workstation from a list of system nodes.
- Obtain a report of the node ID, node roles, and the system time for the local workstation. The report includes the node IDs of all workstations in the DDS/CSF system and the connection status of each node with the acting primary distributor.
- Start programs on local or remote workstations.
- Stop DDS on local or remote workstations.
- Perform a dump of DDS on local or remote workstations.

These administrative functions can be performed from any workstation in the DDS system. DDS must be running on the local workstation and on all of the

workstations that are specified as parameters in order for a command to be successful.

See “[Starting the Node Control Utility](#)” on page 16 for detailed descriptions of each of the Node Control functions. See “[FDSADMIN](#)” on page 25 for information about using the **FDSADMIN** command line interface to the Node Control Utility.

FDSADMIN

Purpose

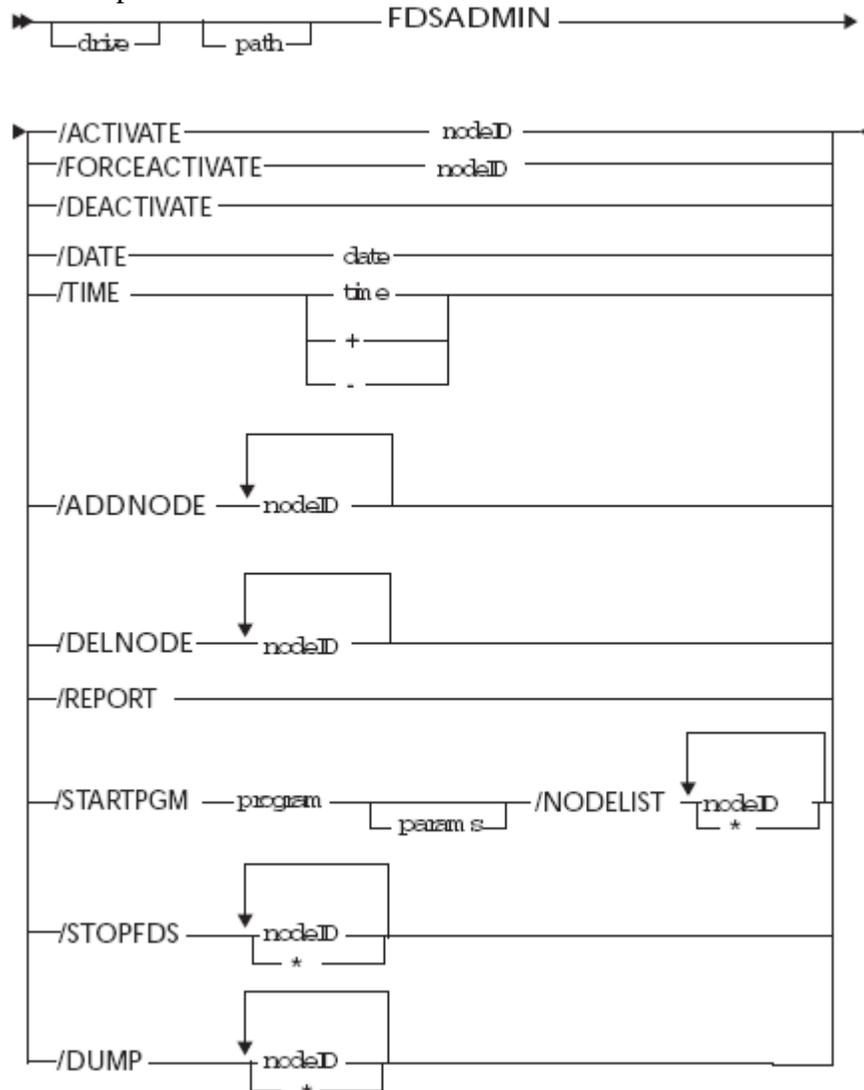
Provides a command line interface to the Node Control functions in Administrator mode.

Syntax

See “[Syntax Conventions](#)” on page vi for information about reading syntax diagrams.

You can obtain help for each of the functions provided by **FDSADMIN** by typing **FDSADMIN ?** and pressing **Enter**.

For each of the parameters shown in the syntax diagram, you can specify either a forward slash (/) or a dash (-) as the delimiter. For example, **FDSADMIN /DEACTIVATE** could also be specified as **FDSADMIN -DEACTIVATE**.



Parameters

ACTIVATE

Activates as the acting primary distributor either the workstation that is the configured primary distributor or the workstation that is the configured backup distributor. The node ID must specify either the configured backup distributor or the configured primary distributor.

This command can activate only those workstations where data distribution is configured and where the workstation is configured as the configured primary distributor or the configured backup distributor.

FORCEACTIVATE

Activates as the acting primary distributor either the workstation that is the configured primary distributor or the workstation that is the configured backup distributor. This command forces the workstation to activate as the acting primary distributor, even if its files are not reconciled with the previous acting primary distributor. This command can result in data being lost if the files on the workstation to be activated are older than the files on the previous acting primary distributor. The node ID must specify either the configured backup distributor or the configured primary distributor. This command can activate only those workstations where data distribution is configured and where the workstation is configured as the configured primary distributor or the configured backup distributor.

DEACTIVATE

Deactivates the current acting primary distributor. This must occur before the acting backup distributor can be activated as the acting primary distributor unless the current acting primary distributor is no longer communicating with the DDS system. This command can only deactivate workstations where data distribution is configured and where the workstation is acting in the role of acting primary distributor.

DATE

Updates the date on the acting primary distributor and synchronizes the date with all workstations on the DDS system. If the workstation is no longer communicating with the acting primary distributor, only that workstation's date is updated.

The date format is similar to that configured for the Windows operating system. The format depends on the information specified when installing Windows. For example, if U.S. English was selected during the installation of Windows, the date format would be mm/dd/yyyy where mm represents month, dd represents day, and yyyy represents year. If another country was selected, the format follows the conventions for that country.

TIME

Updates the time on the acting primary distributor and synchronizes the time with all workstations on the DDS system. If the workstation is no longer communicating with the acting primary distributor, only that workstation's time is updated. Variables for this parameter are:

time Specifies the time to be set. Only hours and minutes are accepted, and they must be expressed in a 24-hour format.

The time format is the same as that configured for the Windows operating system.

+ **(Plus)**

Increments the time by one hour. If the hour increments the time from before midnight to after midnight, the date is also incremented.

– **(Minus)**

Decrements the time by one hour. If the hour decrements the time from after midnight to before midnight, the date is also decremented.

ADDNODE

Adds the specified node IDs to a list of known node IDs maintained by DDS. This allows you to add a node ID to the list even if the workstation has never been connected to the DDS system.

DELNODE

Deletes the specified workstations from the list of node IDs maintained by DDS.

REPORT

Displays a node status report. The report includes the node ID, node roles, and the system time for the local workstation, as well as the node IDs of all workstations in the DDS system and each node's connection status with the acting primary distributor.

The acting primary distributor maintains the list of workstations in your DDS system and the connection status for those workstations. Therefore, the Node Control Report can be viewed only if the acting primary distributor is online and active. Otherwise, a message is returned stating that the status of the workstation is not available.

Note: Nodes may not immediately be detected as inactive. The maximum amount of time between a node's becoming inactive and it's being detected as inactive depends on the type of transport layer being used.

For TCP/IP, this time period is specified by the configuration keyword `IPCHeartbeatInterval`. The default value is 30 seconds.

STARTPGM

Starts a specified program on one or more workstations.

Note: If you need to start an application on a Windows shared drive or an NFS mounted drive, refer to the appropriate Windows or NFS application documentation for configuration requirements. The shared drive must be configured to be accessible by DDS.

Variables for this parameter are:

program

Specifies the name of the program to be started. The program name must be the name of a valid program that exists on the specified workstation.

The program must also be located in a directory that is in the PATH= statement in the specified workstation's system configuration.

Be explicit when specifying the program name. Include the file extension if one exists, so that this utility will start the correct program. For example, if you specify filename MAJOR and there are two programs on the target workstation with that name, one with an .EXE extension and one with no extension, the program MAJOR with no file extension will be started by this utility.

params

Specifies the parameter list for the program to be started. If more than one parameter is to be passed to the program to be started, this value must be enclosed in double quotation marks (for example, "xxx yyy").

NODELIST

Specifies the list of node IDs where a program is to be started.

STOPFDS

Stops DDS on the specified workstations.

DUMP

Forces a DDS dump on the specified workstations. This does not stop DDS.

Remarks

FDSADMIN can be run on a workstation only if DDS/CSF is also running on the workstation. The drive and path variables in the syntax diagram are the drive and subdirectory path where the DDS executable files are installed.

The node ID variable in the syntax diagram specifies the identifier of a workstation. A node ID is required for the following parameters:

- **ACTIVATE**
- **ADDNODE**
- **DELNODE**
- **DUMP**
- **FORCEACTIVATE**
- **STARTPGM**
- **STOPFDS**

Examples

The following examples show commands that you can enter from the command line. These examples assume that the DDS executable files are installed in a directory called \FDS\BIN on a workstation's C drive.

Note: Specifying the drive and directory before the command is optional if one of the following is true:

- the directory where the DDS executable files are installed is in the PATH= statement of a workstation's system configuration file
- the command is entered from the directory where the DDS executable files are installed.

1. To deactivate a workstation as the acting primary distributor:
C:\FDS\BIN\FDSADMIN /DEACTIVATE
2. To activate a workstation with a node ID of mynode as the acting primary distributor:
C:\FDS\BIN\FDSADMIN /ACTIVATE mynode

If another workstation is already the acting primary distributor, that workstation first must be deactivated as the acting primary distributor for this command to be successful.

Note: This command can succeed only when Data Distribution is configured on the specified workstation and the specified workstation is also configured as a configured primary distributor or a configured backup distributor.

3. To set the date on all active workstations to July 26, 1996: C:\FDS\BIN\FDSADMIN /DATE 7/26/1996
4. To set the time on all active workstations to 1:00 p.m.:
C:\FDS\BIN\FDSADMIN /TIME 13:00
5. To add workstations with node IDs of mynode1 and mynode2 to the node status list:
C:\FDS\BIN\FDSADMIN /ADDNODE mynode1 mynode2

The number of nodes that can be added at one time is limited only by the number of characters that can be entered on the command line. The syntax for this command also applies to the **FDSADMIN /DELNODE** command.

6. To view the node status report:
C:\FDS\BIN\FDSADMIN /REPORT
7. To immediately stop DDS at a workstation with a node ID of mynode:
C:\FDS\BIN\FDSADMIN /STOPFDS mynode
8. To immediately stop DDS at two workstations with node IDs of mynode1 and mynode2:
C:\FDS\BIN\FDSADMIN /STOPFDS mynode1 mynode2
9. To immediately stop DDS at all workstations:
C:\FDS\BIN\FDSADMIN /STOPFDS *

Note: The syntax for this command also applies to the **FDSADMIN /DUMP** command.

10. To start a program called MYEXE.EXE with one parameter (myparam) on a workstation with a node ID of mynode:
C:\FDS\BIN\FDSADMIN /STARTPGM myexe.exe myparam /NODELIST mynode
11. To start a program called MYEXE.EXE with two parameters (myparam and parm2) on a workstation with a node ID of mynode:
C:\FDS\BIN\FDSADMIN /STARTPGM myexe.exe "myparam parm2"
/NODELIST mynode
12. To start a program called MYEXE.EXE with no parameters on workstations with node IDs of mynode, yournode, and node1:
C:\FDS\BIN\FDSADMIN /STARTPGM myexe.exe /NODELIST mynode yournode node1
13. To start a program called MYEXE.EXE with no parameters on all workstations:
C:\FDS\BIN\FDSADMIN /STARTPGM myexe.exe /NODELIST *
14. To start a batch file called MYBAT.BAT with one parameter (myparam) on a workstation with a node ID of mynode:
C:\FDS\BIN\FDSADMIN /STARTPGM cmd.exe "/c mybat.bat myparam"
/NODELIST mynode

The Windows program CMD.EXE must be used to start batch files using the FDSADMIN /STARTPGM command. Either /c or /k must be specified before the batch file name in the parameter list for this command to be successful.

- /c Specifies that the Windows window be closed after the batch file stops.
- /k Specifies that the Windows window be kept open after the batch file stops, provided the batch file does not end with an **EXIT** command.

For more information about these parameters, refer to the appropriate Windows documentation about the **CMD** program. You can also type **HELP CMD** from the command line to view information online about the **CMD** program.

Return Codes

- 0** Operation was successful
- 1** DDS is not running
- 3** Command canceled by user
- 4** Internal error
- 5** Operating System Error
- 6** Missing or incorrect keyword entered
- 7** Incorrect argument entered
- 8** Files are not reconciled
- 9** Incorrect number or arguments entered
- 10** Not enough memory available to process request
- 11** Failed to complete request on one or more nodes
- 12** Report information incomplete
- 21** A workstation already is the acting primary distributor
- 99** Command canceled by user

Chapter 4. Problem Analysis Utility

You can use the Problem Analysis Utility to perform problem determination and analysis in DDS/CSF. You can also perform the functions available with the Problem Analysis Utility from the command line. See “[Using the Problem Analysis Utility from the Command Line](#)” on page 37 for more information.

The Problem Analysis Utility helps you to collect dump files, configuration files, and other error files used for problem determination and analysis.

DDS/CSF uses the Windows event log service to assist with problem analysis. When an event or error occurs, DDS/CSF writes a record of the event or error in the Windows operating system application log. The error information can then be viewed using the event log viewer provided by Windows. Refer to the appropriate Windows documentation for more information about the event log service. You can use the Problem Analysis Utility to collect this error information along with other types of files used in diagnosing problems.

The Problem Analysis Utility has an interactive panel on which you can select the types of files you want to collect and the location where you want the information to be saved. The utility then compresses the files to a temporary file in the DDS work directory, and uses the command **FDSCOMP** to write the temporary file **FDSSAVE.FIL**. This file is placed in the subdirectory **FDSSAVE_x**, where *x* is a number from 0 to 9, based on the number of collections you have specified. You can specify up to 10 collections per disk drive. If you specify more than 10 collections, the oldest collections are overwritten.

You can specify that the subdirectories and files be stored on any existing disk or diskette drive. The resulting diskettes or files can then be sent to your QVS customer support team to determine the nature of the problem.

Starting the Problem Analysis Utility

DDS must already be installed on the workstation before you can start the Problem Analysis Utility. However, DDS does not need to be running on the workstation to use this utility.

To start the Problem Analysis Utility, click **Start** from the taskbar, point to **Programs**, point to **QVS DDS**, and click **Problem Analysis Utility**. You can also type **FDSPADU** and press **Enter** from the command line.

The main panel of the Problem Analysis Utility is displayed with the following menu options:

- DDS Configuration Files
- DDS Dump Files
- Application Function Library Configuration and Option Files

- Windows Event Logs
- Windows Dr. Watson Logs
- Windows Memory Dump
- System Information
- User-Defined Files
- 4690 Controller Services Feature Diagnostic Files

Select the items you want to collect. Then press **Create**. Refer to the individual sections for each item you selected for descriptions of the items and additional instructions.

DDS Configuration Files

The configuration files contain information about your DDS/CSF operating environment. These files are automatically created by the DDS Configuration and Response File Utility when you install and configure DDS on a machine. The files that are collected by the Problem Analysis Utility are:

- FDSCONFIG.CFG
- FDSUSED.CFG
- FDSWS.CFG
- FDSL.NLN

DDS must be installed in order for these files to be collected. In addition, the `FDSDIR=x:\xxx` statement must be defined in the system configuration, where `x:\xxx` is the drive and path where DDS is installed. The configuration files are located in the `CONFIG` subdirectory in the DDS installation directory.

Instructions for Problem Analysis Utility

After you select **Create** from the main menu, the utility collects the files at the destination drive you selected. There are no additional steps.

DDS Dump Files

The DDS dump files are created by DDS whenever an unrecoverable error condition occurs. These files contain information that shows the state of DDS when the error occurred. The files are named `FDSCxxx.DMP`, where `xxx` is a unique numeric identifier for the file. These files are located in the `DDS WORK` directory.

Instructions for Problem Analysis Utility

After you select **Create** from the main menu, the Select DDS Dump Files panel is displayed. This panel allows you to specify the search criteria used by the utility to generate a list of process dumps. You can then select any or all of the dumps in this list to be collected.

1. Specify a start date and end date. The start and end date fields are used to limit the dumps included in the list based on the date that the dump was created. You can optionally specify that dumps previously collected by the utility be omitted from the list.
2. Select Search. The Problem Analysis Utility searches for files that were created within the date range you indicated, and displays them.
3. Select the files you want to collect.
4. Select Accept.

The utility collects the files at the destination drive you selected.

Application Function Library Configuration and Option Files

The IBM StorePlace Application Function Library (AFL) configuration and options files contain information about your AFL operating environment, if the AFL is installed. These files are automatically created by the AFL when you install, configure, and personalize AFL on a machine.

IBM StorePlace Application Function Library must be installed for these files to be collected, and the statement `FST_STORE_DIRECTORY=x:\xxx` must be defined in your system configuration, where `x:\xxx` is the disk drive and subdirectory where the AFL configuration and options files are located.

Instructions for Problem Analysis Utility

After you select Create from the main menu, the utility collects the files at the destination drive you selected. There are no additional steps.

Windows Event Logs

The event logs contain information from both DDS/CSF and other products about events and errors. The information in the event logs shows the sequence of events that precede a failure.

The event logs are viewable using the Windows Event Viewer. For information on using the Windows Event Viewer and the information contained in the event logs, refer to your Windows documentation.

Three files are collected, one for each of the following Windows event logs:

- Application log (APPROG.TXT). This log contains event information from applications. The DDS/CSF events are located in this file.
- System log (SYSLOG.TXT). This log contains event information from the Windows operating system and device drivers. File System Interface (FSI) events are also contained in this log.
- Security log (SECLOG.TXT). This log contains security event information.

Output files from these logs are created by the Problem Analysis Utility and collected.

Instructions for Problem Analysis Utility

After you select **Create** from the main menu, the utility collects the files at the destination drive you selected. There are no additional steps.

Windows Dr. Watson Logs

Dr. Watson logs contain information about the state of a process when an error occurs that causes the process to trap. The information stored in the logs is cumulative; the logs are updated whenever a process traps. When an unrecoverable error occurs within DDS/CSF, the Dr. Watson log files are updated with the error information.

There are two Dr. Watson log files:

DRWTSN32.LOG

This file is located in the directory specified by the LogFilePath key value in the \HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\DrWatson object in the registry.

USER.DMP

This file is located in the path specified by the CrashDumpFile key value in the \HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\DrWatson object in the registry. The default file name is USER.DMP, but another file name can be specified.

The Dr. Watson logs will not be created unless the CreateCrashDump key value in the \HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\DrWatson object is set to TRUE. Distributed Data Services automatically sets this key value to TRUE during installation.

Instructions for Problem Analysis Utility

After you select **Create** from the main menu, the utility collects the files at the destination drive you selected. There are no additional steps.

Windows Memory Dump

The Windows memory dump file contains the information stored in memory when a serious error occurs. This file will only be created when CrashDumpEnabled key value in the \HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\CrashControl object is set to TRUE. Distributed Data Services automatically sets this key value to TRUE at installation.

The file path for the memory dump file is specified by the DumpFile key value, and the default file name is MEMORY.DMP. In addition, the following .SYS files are collected:

FDSFIFIL.SYS, FDSFIFUN.SYS, FDSFIREC.SYS and if CSF is installed, FDSLANT.SYS.

Because the size of this file can be as large as the amount of memory installed in the workstation, it is not recommended that you collect this file unless requested to do so by your QVS support representative.

Instructions for Problem Analysis Utility

After you select **Create** from the main menu, the utility collects the files at the destination drive you selected. There are no additional steps.

System Information

The system information consists of two files:

- Output from the Windows **SET** command
The **SET** output file, ENV.TXT, contains system environment information for the workstation. The Problem Analysis Utility creates the output file by using the **SET** command and redirecting the output to a file in the Distributed Data Services WORK directory.
- Output from the FDSSYS utility
The FDSSYS output file, SYSINFO.TXT, contains certain hardware and operating system information for the workstation. It is stored in the current directory.

Instructions for Problem Analysis Utility

After you select **Create** from the main menu, the utility collects the files at the destination drive you selected. There are no additional steps.

User-Defined Files

The Problem Analysis Utility allows you to choose any additional files that you would like collected with the other error information, such as output from other diagnostic utilities. Be careful not to specify any files that contain information that is confidential to your company.

Instructions for Problem Analysis Utility

After you select **Create** from the main menu, the Enter User-Defined Files panel is displayed.

1. Either type the name of the file you want to collect, or select a disk drive and search directory from the list and then select the file.
2. Select **Add**.

3. You can select more files by repeating the previous two steps for each file you want to select.
4. When you have finished selecting files, select **Accept**. The Enter User-Defined Files panel is displayed again, with the files you selected added to the list. You can remove files by selecting a file name and selecting **Remove**. Repeat this step for each file you want to remove.
5. When you have added all the files you want to add, select **Accept**.

The utility collects the files you selected.

Attention: When multiple files are collected in different directories, but with the same file name, only the last one decompressed will be saved. Others that have the same file name will be overwritten.

4690 Controller Services Feature Diagnostic Files

The Problem Analysis Utility allows you to collect files used to diagnose 4690 Controller Services problems when 4690 Controller Services is installed on the workstation. The primary files that are collected are:

ADXCSLTF.DAT

This is the 4690 terminal dump file. This file is located in the x:\ADX_SDT1 subdirectory, where x is the first valid drive where an \ADX_SDT1 directory is found.

ADXCSNRF.DAT

This is the 4690 event-log report file. This file is also located in the x:\ADX_SDT1 subdirectory, where x is the first valid drive where an \ADX_SDT1 directory is found.

xxxxxxx.BSX

These are 4690 symbols files, where xxxxxxxx is any name corresponding to a file with a .286; extension. These files are located in the x:\ADX_SPGM subdirectory, where x is the first valid drive where an \ADX_SPGM directory is found. The Problem Analysis Utility collects all files in the \ADX_SPGM directory with a .BSX; extension.

Other files related to the 4690 Controller Services Feature are also collected.

Refer to the Create Problem Analysis Diskette section in the *IBM 4690 Store Systems: Messages Guide* for more information about these files.

Instructions for Problem Analysis Utility

After you select Create from the main menu, the utility collects the files at the destination drive you selected. There are no additional steps.

Using the Problem Analysis Utility from the Command Line

The Problem Analysis Command provides a command line interface for performing the same functions as the Problem Analysis Utility. Functions that can be performed are:

- Collect all files needed to diagnose DDS, CSF, or AFL problems
- Collect only the files needed to diagnose DDS problems
- Collect only the files needed to diagnose CSF problems
- Collect only the files needed to diagnose AFL problems
- Collect the Windows memory dump
- Collect a specific file not included in the groups of files listed previously.
- Collect only dump files created on or after a specified date.
- Specify the disk drive on which to save the files.
- Collect a group of files listed in the FDSPADC.LST file, which is located in the DDS working directory.

DDS/CSF does not need be running on the local workstation in order for this command to be successful.

FDSPADC

Purpose

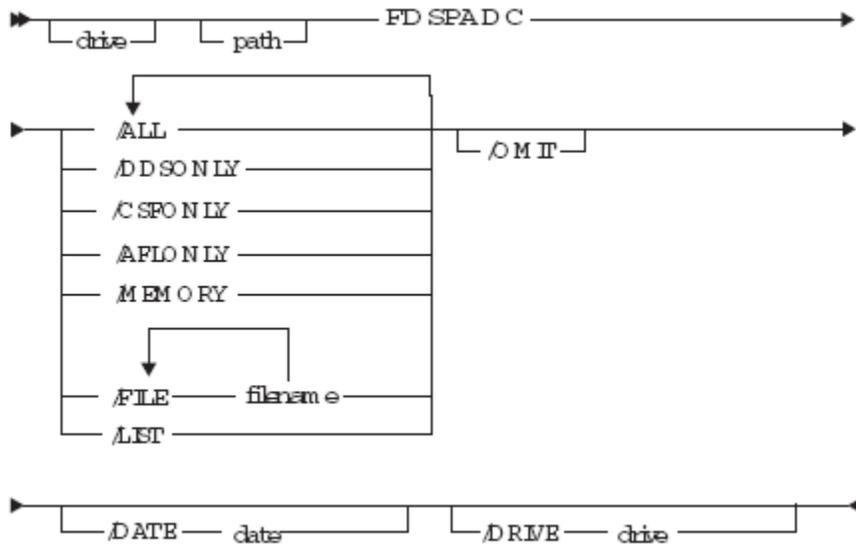
Provides a command line interface to the Problem Analysis Utility functions.

You can obtain help for each of the functions provided by **FDSPADC** by typing **FDSPADC ?** and pressing **Enter**.

Syntax

See "[Syntax Conventions](#)" on page vi for information about reading syntax diagrams.

For each of the keywords shown in the syntax diagram, you can specify either a forward slash (/) or a dash (-) as the delimiter. For example, **FDSPADC /ALL /DRIVE a:** could also be specified as **FDSPADC -ALL -DRIVE a:**.



Parameters

ALL

Collects all files needed to diagnose a problem for DDS, CSF, or the AFL. The files collected are: DDS configuration files, DDS dump files, the AFL files listed in FDSPADU.LST if configured, the CSF diagnostic files if configured, Windows event logs, Windows Dr. Watson logs, system information files, and files listed in the FDSPADC.LST file located in the DDS WORK directory. Does not collect memory dump files.

DDSONLY

Collects the files needed to diagnose a problem for DDS. The files collected are: DDS configuration files, DDS dump files, Windows event logs, Windows Dr. Watson logs, and system information files.

CSFONLY

Collects the files needed to diagnose a problem for the DDS Controller Services Feature. The files collected are: DDS configuration files, CSF diagnostic files, and system information files.

AFLONLY

Collects the files needed to diagnose a problem for the Application Function Library. The files collected are: files listed in the FDSPADU.LST file, which is stored in the directory indicated by the FST_STORE_DIRECTORY environment variable; Windows event logs; Windows Dr. Watson logs; and system information files.

MEMORY

Collects the Windows memory dump file. Because the size of this file can be as large as the amount of memory on a workstation, it is not recommended that you collect this file unless directed to do so by an QVS support representative.

FILE Collects the file specified after this keyword. This keyword can be specified more than once.

LIST Collects the files specified in the FDSPADC.LST file located in the DDS/CSF WORK directory.

If you are going to specify the **LIST** parameter, you must first create the file FDSPADC.LST. Use any ASCII editor to create the file, adding the file name of the file to be collected (add one file name per line).

Notes:

1. To ensure that you collect the file name you want, you should specify a fully qualified file name (include the drive and the directory path).
2. You cannot use file names that contain spaces within the name.

DATE

Collects only dump files that were created on or after the date specified. The date must be in the format defined for the operating system by the user. This keyword is ignored for files that are not dump files.

OMIT

Omits dump files that have been previously collected. This keyword is ignored for files that are not dump files.

DRIVE

Specifies the disk drive where the files are to be saved. If this keyword is not specified, the drive on which DDS is currently running is used.

Remarks

DDS/CSF does not need to be running for this command to be successful.

The drive and path variables in the syntax diagram are the drive and subdirectory path where the DDS/CSF executable files are installed. The path must include the subdirectory.

When you use **FDSPADC**, it compresses the files to a temporary file in the DDS/CSF work directory, and uses the command **FDSCOMP** to write the temporary file **FDSSAVE.FIL**. This file is placed in the subdirectory **FDSSAVE_x**, where **x** is a number from 0 to 9, based on the number of collections you have specified. You can specify up to 10 collections per disk drive. If you specify more than 10 collections, the oldest collections are overwritten.

Duplicate keywords are ignored. If combinations of keywords are specified, such as **/ALL** and **/DDSONLY**, **FDSPADC** will collect diagnostic files for both. Files that are collected for both instances of a keyword will be collected only once.

Examples

The following examples assume that the DDS executable files are installed in the directory called **\FDS\BIN** on a workstation's C drive. Specifying the drive and directory before the command is optional if one of the following is true:

- The directory where the DDS executable files are installed is in the **PATH=** statement of a workstation's system configuration.
- The command is entered from the directory where the DDS executable files are installed.
- To collect all diagnostic files and save them to drive A:
`C:\FDS\BIN\FDSPADC /ALL /DRIVE a:`
- To collect the diagnostic files for the CSF and the files listed in the **FDSPADC.LST** file to drive C:
`C:\FDS\BIN\FDSPADC /CSFONLY /LIST /DRIVE c:`
- To collect the AFL diagnostic files and two other files in the current directory called **MYFILE.FIL** and **MYFILE2.FIL**:
`C:\FDS\BIN\FDSPADC /AFLONLY /FILE myfile1.fil /FILE myfile2.fil`
- To collect the DDS diagnostic files, and only include dump files created after November 25, 1996 and omit dump files that have been previously collected:
`C:\FDS\BIN\FDSPADC /DDSONLY /DATE 11/25/1996 /OMIT`

Return Codes

- | | |
|----------|--------------------------|
| 0 | Operation was successful |
| 3 | Command canceled by user |
| 4 | Internal error |
| 5 | Operating system error |

- 6** Missing or incorrect keyword entered
- 7** Incorrect argument entered
- 9** Incorrect number of arguments entered
- 10** Not enough memory available to process command
- 15** DDS is not configured correctly
- 19** Disk full
- 20** Another copy of FDSPADC is already running
- 27** Initialization error
- 37** Path not found
- 38** Cannot access a drive or file
- 41** File not found
- 48** AFL or CSF is not installed
- 99** Command canceled by user

Chapter 5. Data Distribution Utility

The Data Distribution Utility enables you to manage data distribution for workstations associated with your DDS system. You can specify what data is distributed and where it is distributed.

Most of the Data Distribution Utility functions can be initiated from a panel-driven graphical user interface or from the command line. (You cannot use Exception Management from the command line.) See “[Starting the Data Distribution Utility](#)” for more information about using the graphical user interface. See “[Using the Data Distribution Utility from a Command Line](#)” on page 44 for more information about taking action from the command line.

Note: You can use the Data Distribution Utility from the primary distributor only.

Starting the Data Distribution Utility

To start the Data Distribution Utility, click **Start** on the taskbar on the primary distributor, point to **Programs**, point to **DDS**, and click **Data Distribution Utility**. You can also start the utility from a command line by typing **FDSDDU** and pressing **Enter**.

Note: DDS must already be running on the workstation before you can start the Data Distribution Utility.

Four menu bar options are available from the main Data Distribution Utility panel:

- Broadcast Domain Management, which allows you to create, delete, and change a workstation in a broadcast domain
- Directory Management, which allows you to set distribution using a file name or a logical name, or to show distributed files
- Exception Management, which allows you to manage online, offline, reconciling, or all workstations
- Help, which provides online help for panels

Broadcast Domain Management

A *broadcast domain* is a group of workstations that are identified by a broadcast domain name. The broadcast-domain name is used like a mailing list by the Data Distribution component to send file updates.

Select **Broadcast Domain Management** from the main panel. The following options are displayed:

- Create a Broadcast Domain
- Delete a Broadcast Domain
- Change a Workstation in Broadcast Domain

For information about using the Broadcast Domain Management service from a command line, see “[FDSBDOMN](#)” on page 49.

Create a Broadcast Domain

To create a broadcast domain:

1. Select **Create a Broadcast Domain...** from the Broadcast Domain Management options pull-down.
2. Type the name of the broadcast domain to be created.
3. Select one or more node IDs of the workstations to be added to the broadcast domain. If you do not select any workstations, an empty broadcast domain is created.
4. Select **Accept**.

The broadcast domain is created with the name and workstations you indicated.

Delete a Broadcast Domain

To delete a broadcast domain:

1. Select **Delete a Broadcast Domain...** from the Broadcast Domain Management options pull-down.
2. Select the broadcast domain to be deleted.
3. Indicate whether you want to confirm the deletion. A check in the **Confirm delete** check box will cause you to be prompted for confirmation before the broadcast domain is deleted.
4. Select **Accept**.

Change a Workstation in Broadcast Domain

This menu option allows you to add and delete one or more workstations in broadcast domains.

To change a workstation in a broadcast domain:

1. Select **Change a Workstation in Broadcast Domain...** from the Broadcast Domain Management options pull-down.
2. Select the name of the broadcast domain you want to change.
3. Select **Accept**.
4. Select one or more workstations. Workstations that are currently highlighted belong to the broadcast domain. Deselecting a workstation indicates that you want to delete it from the broadcast domain. Selecting a workstation indicates that you want to add it to the broadcast domain.
5. Selecting **Undo** resets the highlighted workstations to the default selections.
6. Select **Accept**.

The actions you requested are processed.

Directory Management

Directory management allows you to manage distribution at the file level or at the subdirectory level.

Note: DDS/CSF supports the distribution of files up to a maximum size of 4 GB.

Each distributed file or distributed subdirectory is associated with a group of attributes that control how often distribution occurs, where the files or subdirectories are distributed, and the scope of the distribution. The attributes are:

Distribution name

Indicates a valid file or subdirectory name. The file or subdirectory must exist on the primary distributor, and can be distributed only to either a broadcast domain or mirrored domain.

Directory indicator

Indicates whether the distribution name is a file or a subdirectory.

Domain type

Specifies the nature of the destination of the file or subdirectory. A file or subdirectory can have one of these domain types:

This Type: Causes This Action:

Mirrored

The file or subdirectory is distributed only to the acting backup distributor.

Broadcast

The file or subdirectory is distributed to all workstations defined by the name specified by the distribution domain name.

Local

The file or subdirectory is deleted from the distribution directory and all image copies are deleted. The file or subdirectory on the primary distributor is the only remaining copy.

Distribution domain name

Indicates the name of a broadcast domain. Broadcast domains contain the names of the workstations where the associated files or subdirectories are to be distributed. You can define broadcast domain names using Broadcast Domain Management. See “[Broadcast Domain Management](#)” on page 39 for more information.

Distribution frequency

Specifies when files or subdirectories are distributed. Files or subdirectories can be:
Distributed on update
Distributed on close

Refer to *QVS Distributed Data Services/Controller Services Feature for Windows Programming Guide* for an explanation of distribution frequency.

Scope qualifier

Specifies the extent of distribution in a subdirectory. You can distribute only the files in the subdirectory or you can distribute all the files and subdirectories, including any file that is located in any subdirectory listed under the subdirectory specified by the distribution name. This attribute is ignored if the distribution name is specified as a file.

To use the Directory Management functions, select **Directory Management** from the main panel. The following options are displayed:

- Set Distribution Using a File Name
- Set Distribution Using a Logical Name

For information about using the Distribution Directory Management service from a command line, see “[FDSSETD](#)” on page 54.

Set Distribution Using a File Name

To set distribution using a file name:

1. Select **Set Distribution Using a File Name...** from the Directory Management options pull-down.
2. Type the name of the file or subdirectory to be distributed, or select an existing file or subdirectory from the list. The file or subdirectory must already exist before you can add it to the distribution directory.
3. Select **Accept**. The Set Distribution panel is displayed.
4. Select a broadcast domain or domain type for the file or subdirectory you indicated.
5. Indicate the distribution frequency for the file or subdirectory. See the *QVS Distributed Data Services/Controller Services Feature for Windows Programming Guide* for an explanation of distribution frequency.
Selecting **Undo** resets the broadcast domain or domain type, and the distribution frequency to the default selections.
6. If you selected a directory, specify the scope qualifier.
7. To redistribute existing files and subdirectories that have not been changed, select **Force distribution**.
8. To create an entry in the distributed directory or to change the distribution characteristics of an existing entry, select **Accept**.

Set Distribution Using a Logical Name

To set distribution using a logical name:

1. Select **Set Distribution Using a Logical Name...** from the Directory Management options pull-down.
2. Type the logical name of a file or subdirectory. The logical name must resolve to a file name or subdirectory that already exists. Logical names are created during the initial configuration of DDS. See “[Logical Names and Role Names](#)” on page 2 for a description of logical names.

Note: DDS supports the distribution of files up to a maximum size of 4,294,967,294 bytes.

3. Select **Accept**.
4. Select a broadcast domain or domain type for the file or subdirectory you indicated.
5. Indicate the distribution frequency for the file or subdirectory. Refer to *QVS Distributed Data Services/Controller Services Feature for Windows Programming Guide* for an explanation of distribution frequency.

Selecting **Undo** resets the broadcast domain or domain type and the distribution frequency to the default selections.

6. If you selected a directory, specify the scope qualifier.
7. To redistribute existing files and subdirectories that have not been changed, select **Force distribution**.
8. To create an entry in the distributed directory or to change the distribution characteristics of an existing entry, select **Accept**.

Note: To show distributed files, select **Show Distributed Files...** from the Directory Management Options pulldown. Select **Refresh** to see the latest information. You can save this list to a file by selecting **Save Report**.

Showing Which Files are Distributed

You can show which files and directories are distributed in the following manner.

1. Select **Show Distributed Files...** from the Directory Management options pulldown.
2. The Show Distributed Files panel is displayed, showing all the files and directories that are distributed, in alphabetical order. Distributed directory names are preceded by a plus (+) symbol. To display the contents of a directory, either double-click on its name or highlight it and press **Enter**.
3. To save the list of distributed files and directories to a file, select **Save Report**.
4. To refresh the list, select **Refresh**

Exception Management

Use Exception Management to display information about workstations within a broadcast domain that have unreconciled files. Files can be unreconciled for a variety of reasons, including a failing or full disk or interruption of the distribution of a file.

When a file is not reconciled, the workstation is out of synchronization with the primary distributor even though the primary distributor continuously attempts to reconcile the file.

Note: The acting primary distributor is never displayed with any of the Exception Management options.

To display information about workstations, select Exception Management from the main panel. The following options are displayed:

- **Manage Online Workstations**
Use this option to display all workstations within any broadcast domain that are communicating with the primary distributor. If you have not defined a broadcast domain, then only the acting backup distributor, if you have one, is displayed.
- **Manage Offline Workstations**
Use this option to display only those workstations within any broadcast domain that are not communicating with the primary distributor.
- **Manage Reconciling Workstations**
Use this option to display only those workstations within any broadcast domain that are currently being reconciled.
- **Manage All Workstations**
Use this option to display all workstations within any broadcast domain, regardless of their status. If you have not defined a broadcast domain, then only the acting backup distributor, if you have one, is displayed.

The procedures for exception management are the same for each group of workstations:

1. Select the group of workstations you want to manage from the Exception Management options pull-down (online, offline, reconciling, or all).
2. Select one or more workstations. You can choose all the workstations by selecting **Select All** or choose no workstations by selecting **Deselect All**.
3. Select **Accept**.

All files that have missed an update or failed to reconcile are displayed. The description of the file contains the following information:

- Node ID
- Status of the file (descriptions for each status are in the online help for this list box)
- Error return code (descriptions for each return code are in the online help for this list box and in the *QVS Distributed Data Services/Controller Services Feature for Windows Programming Guide*)
- Drive, path, and name of this file

Note: The Exception Management panel will display the real path name for a file or directory, regardless of whether or not a symbolic link was used to set the file or directory to be distributed.

4. In order to clear a file exception, you can either **force distribution** of a file or you can **remove an exception**. Choose **force distribution** if you want to use the acting primary workstation copy of the file as master. Choose **remove an exception** if you do not wish to write over the file copy on the acting backup workstation.

To force distribution:

- a. Correct the cause of the exception at the workstation, if possible, or report the problem to your store programmer.

- b. After the cause of the exception is corrected, select the file on this panel that corresponds to the exception you have fixed.
- c. Select Force distribution. This forces a reconciliation of the file using the acting primary distributor copy as the master.

To remove an exception:

To remove an exception from the list, select the file you want to remove and then select **Accept**. The file is removed from the exception list. You should choose this option only if you do not want to overwrite the file copy on the acting backup workstation.

Using the Data Distribution Utility from a Command Line

All of the functions for Broadcast Domain Management and Directory Management can also be initiated from a command line.

These functions can be performed only from the primary distributor. DDS must be running on the primary distributor in order for a command to be successful.

See “[FDSBDMN](#)” on page 49, “[FDSGETD](#)” on page 52 and “[FDSSETD](#)” on page 54 for information about using the **FDSBDMN**, **FDSGETD**, and **FDSSETD** command line interfaces to the Data Distribution Utility.

FDSBDOMN

Purpose

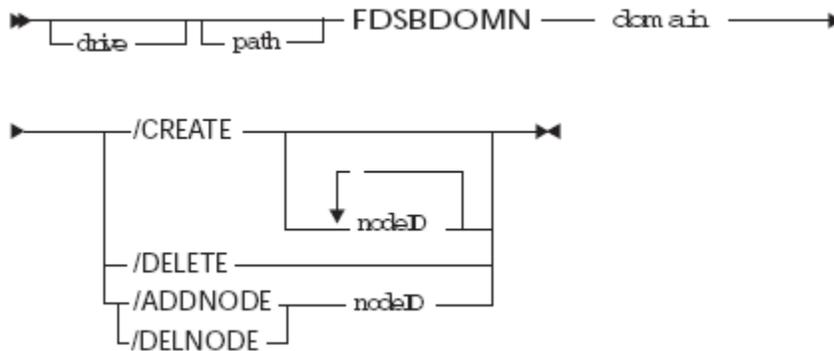
This command allows you to use the Broadcast Domain Management service from a command line. Use this command to create or delete a broadcast domain, or add or delete workstations from a broadcast domain.

Obtain help for each of the functions provided by **FDSBDOMN** by typing **FDSBDOMN ?** and pressing **Enter**.

Syntax

See “[Syntax Conventions](#)” on page vi for information about reading syntax diagrams.

For each of the keywords shown in the syntax diagram, you can specify either a forward slash (/) or a dash (-) as the delimiter. For example, **FDSBDOMN domain1 /CREATE** could also be specified as **FDSBDOMN domain1 -CREATE**.



Parameters

domain

Specifies the broadcast domain name that is to be created, deleted, or modified. This parameter is required.

CREATE, DELETE, ADDNODE, DELNODE

You must specify one of these parameters with the **FDSBDOMN** command. You cannot specify more than one.

CREATE

Create a broadcast domain.

DELETE

Delete a broadcast domain.

ADDNODE

Add a workstation to an existing broadcast domain.

DELNODE

Delete a workstation from an existing broadcast domain.

nodeID

Specifies the identifier of the workstation. This parameter is required only for the ADDNODE and DELNODE parameters.

Remarks

This command line program can be initiated only from the primary distributor. Initiating the command from any workstation other than the primary distributor causes the command to fail and return a non-zero return code.

Examples

1. To create a broadcast domain named domain1:
FDSBDMN *domain1* /CREATE
2. To delete a broadcast domain named domain1:
FDSBDMN *domain1* /DELETE
3. To add a workstation called wks1 to a broadcast domain named domain1:
FDSBDMN *domain1* /ADDNODE *wks1*
4. To delete a workstation called wks1 from a broadcast domain named domain1:
FDSBDMN *domain1* /DELNODE *wks1*

Return Codes

- | | |
|-----------|--------------------------------------------------------|
| 0 | Operation completed successfully |
| 1 | DS not running |
| 3 | Command canceled by user |
| 6 | Missing or incorrect keyword entered |
| 7 | One of the arguments entered is not correct |
| 9 | Incorrect number of arguments entered |
| 10 | Not enough memory available to process request |
| 17 | Workstation you indicated was not found |
| 34 | Data Distribution is not configured |
| 35 | Command was not initiated from the primary distributor |
| 36 | Domain you indicated was not found |

- 39** A broadcast domain already exists, you specified a node ID more than once, or this node ID was assigned to more domains than are currently supported by DDS
- 47** Unexpected error from data distribution
- 99** Command canceled by user

FDSGETD

Purpose

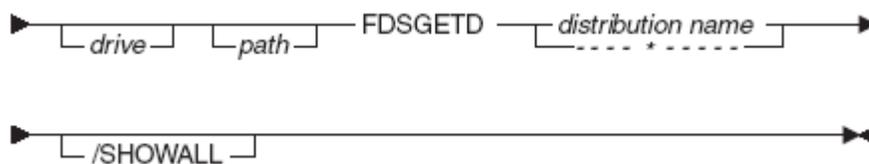
This command allows you to use the Data Distribution directory-management service from a command line. Use this command to determine which files and directories are distributed.

Obtain help for each of the functions provided by **FDSGETD** by typing **FDSGETD ?** and pressing **Enter**.

Format

See “[Syntax Conventions](#)” on page vi for information about reading syntax diagrams.

For each of the keywords shown in the syntax diagram, you can specify either a forward slash (/) or a dash (-) as the delimiter. For example, **FDSGETD d:\mysub /SHOWALL** could also be specified as **FDSGETD d:\mysub -SHOWALL**.



Parameters

distribution name

Indicates a valid file or directory name from which to start the search for all distributed files and directories. Valid file names must include the drive and path specification. To return all files and directories that have been distributed, enter an asterisk (*) as the distribution name.

SHOWALL

For distributed directories with a DIRTREE distribution type, shows all distributed files and subdirectories within directory.

Comments

This command-line program can be initiated only from the acting primary distributor. Initiating the command from any workstation other than the acting primary distributor causes the command to fail and return a non-zero return code.

Examples

1. To show if a file is distributed:

- FDSGETD d:\subdir1\file1
2. To show all the files and directories that are distributed in the d:\subdir1 path:
FDSGETD d:\subdir1 /SHOWALL
 3. To show all files and directories that are distributed:
FDSGETD * /SHOWALL

Return Codes

- | | |
|-----------|-----------------------------------------------------------------------------------|
| 0 | Operation completed successfully |
| 1 | DDS is not running |
| 4 | Internal error occurred |
| 6 | Missing or incorrect keyword entered |
| 9 | Incorrect number of arguments entered |
| 17 | A node could not be found |
| 35 | Command was not initiated from the primary distributor |
| 36 | A domain could not be found |
| 40 | The distribution name specified is not on the controlled drive or does not exist. |

FDSSETD

Purpose

This command allows you to use the Data Distribution directory management service from a command line. You can use this command to set distribution for a file or subdirectory.

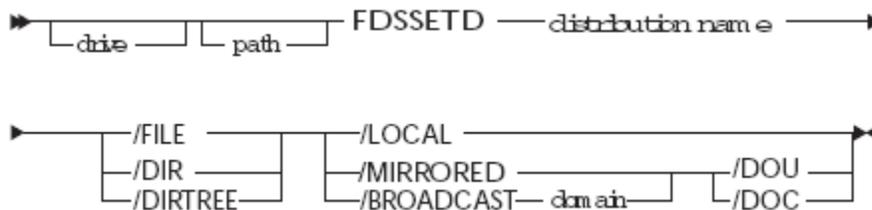
Note: DDS supports the distribution of files up to a maximum size of 4,294,967,294 bytes.

You can obtain help for each of the functions provided by **FDSSETD** by typing **FDSSETD ?** and pressing **Enter**.

Syntax

See "[Syntax Conventions](#)" on page vi for information about reading syntax diagrams.

For each of the keywords shown in the syntax diagram, you can specify either a forward slash (/) or a dash (-) as the delimiter. For example, **FDSSETD d:\mysub /DIR /BROADCAST domain1 /DOU** could also be specified as **FDSSETD d:\mysub -DIR -BROADCAST domain1 -DOU**.



Parameters

distribution name

Indicates a valid file or subdirectory name that is to be distributed. File names must include the drive and path specification to be valid. This value can be a logical name, but it must resolve to a valid file or subdirectory name on the primary distributor. The file or subdirectory must already exist on the primary distributor, and can be distributed to only one broadcast domain or the mirrored domain.

FILE, DIR, DIRTREE

Specifies the directory indicator. That is, the distribution name as a file or subdirectory.

Valid values are:

- FILE** Specifies the distribution name as a file
- DIR** Specifies the distribution name as a subdirectory. Only the files in the subdirectory are distributed.

DIRTREE Specifies the distribution name as a subdirectory. All files and all subdirectories within this subdirectory will be distributed.

DIRTREE

Specifies the distribution name as a subdirectory. All files and all subdirectories within this subdirectory will be distributed.

MIRRORED, BROADCAST, LOCAL

Specifies domain type that indicates the nature of the destination of the file or subdirectory. Valid values are:

MIRRORED

Indicates that the file or subdirectory is distributed only to the acting backup distributor.

BROADCAST

Indicates that the file or subdirectory is distributed to all workstations defined to the name specified by domain as well as the acting backup distributor.

LOCAL

Indicates that the file or subdirectory is not distributed, and all image copies are deleted. The file or subdirectory on the primary distributor is the only remaining copy.

Note: If you specify LOCAL for a subdirectory that was distributed, all image copies of the file within the subdirectory will be deleted. However, the subdirectory itself will not be deleted.

domain

Specifies the distribution domain name of the broadcast domain that contains the names of the workstations where the associated files or subdirectories are to be distributed. The broadcast domain must already exist when this program is initiated.

This parameter is valid only if you specified a domain type of BROADCAST.

DOU, DOC

Specifies the distribution frequency of files or subdirectories that are distributed. Valid values are:

DOU Files or subdirectories are distributed on update.

DOC Files or subdirectories are distributed on close.

Refer to *QVS Distributed Data Services/Controller Services Feature for Windows Programming Guide* for more information about distribution frequency.

Remarks

This command-line program can be initiated only from the primary distributor. Initiating the command from any workstation other than the primary distributor causes the command to fail and return a non-zero return code.

No file or directory in the directories where DDS is installed (including the directory specified by the **WorkDirectory** configuration parameter) can be distributed; nor can the root directory be distributed.

Examples

1. To cause a file named d:\subdir1\MYFILE.DAT to be distributed to the broadcast domain named domain1:
FDSSETD d:\subdir1\myfile.dat /FILE /BROADCAST domain1 /DOU
2. To cause a subdirectory named d:\mysub to be distributed to the broadcast domain named domain1:
FDSSETD d:\mysub /DIR /BROADCAST domain1 /DOU

Return Codes

- | | |
|-----------|-----------------------------------------------------------------------------------------------------------------------|
| 0 | Operation completed successfully |
| 1 | DDS not running |
| 3 | Command canceled by user |
| 6 | Missing or incorrect keyword entered |
| 7 | One of the arguments entered is not correct |
| 9 | Incorrect number of arguments entered |
| 10 | Not enough memory available to process request |
| 34 | Data Distribution is not configured |
| 35 | Command was not initiated from the primary distributor |
| 36 | Domain you indicated was not found |
| 38 | The distribution name specified could not be accessed, and the file or subdirectory is open |
| 39 | The distribution name specified is already being distributed |
| 40 | The distribution file name specified is not on a controlled drive |
| 41 | The distribution file name specified was not found in the directory path specified |
| 42 | The logical name specified in the distribution name is not valid |
| 43 | The logical name specified in the distribution name was not found |
| 44 | The logical name specified in the distribution name was too complex |
| 45 | The distribution name specified was not a local file |
| 46 | The distribution name specified is already distributed or a file in the subdirectory specified is already distributed |
| 47 | Unexpected error from data distribution |
| 49 | This function is not supported on this operating system |
| 50 | Set distribution to LOCAL or MIRRORED failed |
| 51 | Failed to reset distribution to original settings after set distribution to LOCAL or MIRRORED failed |
| 99 | Command canceled by user |

Chapter 6. Keyed File Utility

The Keyed File Utility allows you to create and maintain keyed files. All of the Keyed File Utility functions can be initiated from a panel-driven, graphical-user interface or from the command line. See “[Starting the Keyed File Utility](#)” for more information about using the graphical-user interface. See “[Using the Keyed File Utility From the Command Line](#)” on page 64 for more information about taking action from the command line.

Starting the Keyed File Utility

To start the Keyed File Utility, click Start from the taskbar, point to **Programs**, point to **DDS**, and click **Keyed File Utility**. You can also start this utility by typing **FDSKFUI** and pressing **Enter** from the command line.

Note: DDS must be running on the workstation before you can start this utility.

When you start the Keyed File Utility, the Checkpoint File Found panel might be displayed. See Step 1 in “[Keyed File from Byte Stream File](#)” for more information.

Four menu bar options are available from the main Keyed File Utility panel:

- Create, which allows you to create a keyed file from a byte-stream file, create a byte-stream file from a keyed file, and create an empty keyed file
- Report, which provides statistics for keyed files and chaining statistics for byte-stream files
- Verify, which allows you to verify the integrity of keyed files
- Help, which provides online help for panels

In byte-stream files, data is retrieved by reading from the beginning of the file and continuing until the required information is found. In keyed files, data is located using a key. The key points to the location in the file where the information is stored. For example, a musical cassette tape contains data in byte-stream format, and a musical compact disk contains data in keyed-file format.

Create

Select **Create** from the main panel. The following options are displayed:

- Keyed File from Byte Stream File
- Byte Stream File from Keyed File
- Empty Keyed File

Keyed File from Byte Stream File

You can use an existing byte-stream file as input to create a new keyed file.

To create a new keyed file from an existing byte stream file:

1. Select **Keyed File from Byte Stream File...** from the Create options pull-down.

The Checkpoint File Found panel might be displayed. Checkpoints are performed by the Keyed File Utility at 20-minute intervals. A checkpoint file is created if the keyed file creation process is interrupted. The next time you select the Keyed File from Byte Stream File option, a panel displays the name of the input and output files for which the checkpoint file exists. You can restart the process from the last checkpoint or you can delete the checkpoint file.

Select one of the following options if the Checkpoint File Found panel is displayed:

Delete Delete the checkpoint file.

Note: If the file is small, deleting the file is more efficient. If the file is large, it is more efficient to restart.

Restart Restart the creation of the keyed file from the existing checkpoint.

Ignore Ignore the prompt and keep the checkpoint file. This panel will be displayed again the next time you select the Keyed File from Byte Stream File option or restart the Keyed File Utility.

From the Create Keyed File from Byte Stream File panel, type the name of the existing byte stream file or select it from the list.

2. Type the name for the new keyed file you are creating or select it from the list. If the file already exists, a panel is displayed that prompts you to verify that the file should be overwritten. If you do not want to overwrite the file, select No and you will be returned to the Create Keyed File from Byte Stream File panel to enter a different file name.
3. Type the record length for the file (in the range 1 to 4 092).
4. Type the key length for the file (must be less than or equal to the record length).
5. Type the percentage of extra space to be left available in the new file.
6. Select the hashing method to use for the file. See "[Hashing Algorithms](#)" on page 68 for more information about hashing methods. Based on your input, the system calculates default values for the following data and uses them for creating the new keyed file:
 - Block size
 - Randomizing divisor
 - Chaining threshold

You can select **Change Calculated Defaults** to view or change the default values. The window that is displayed lists the values that have been calculated from the values you indicated at the Create Keyed File from Byte Stream File panel.

You can change the calculated values or you can exit from the panel and return to the Create Keyed File from Byte Stream File panel.

If you change the calculated values, the values on the Create Keyed File from Byte Stream File panel do not change. Similarly, if you have changed the calculated default values, and then change values on the Create Keyed File from Byte Stream File panel, the default values you changed are still in effect.

If you have changed the calculated values and want to recalculate the values based on the values on the Create Keyed File from Byte Stream File panel, select **Use Dynamic Defaults**.

7. Put a check in the **Erase Input File** check box to erase the byte-stream file after the keyed file is created. Do not put a check in the check box to save the byte-stream file.
8. Select **Accept** when all information is complete to create the keyed file. The Create Keyed File Status panel displays the following information:
 - Elapsed time
 - Input file
 - Output file
 - Status messages
 - Percent complete

Messages are displayed to tell you the status of your request. You can cancel the request at any time by selecting **Cancel**. If you cancel the request at this time, a checkpoint may be created.

9. You can browse the status messages and select **Return** when you have finished, or you can save or print the status messages.

Select **Save Status** to save the messages. The Save Request Status panel is displayed and you are prompted to select the output destination as either a file or a printer port:

- To save the status messages to a file, select **File** as the output destination, type the output file name or select it from the list, and select **Accept**. The previous panel is displayed.
- To print the status messages, select one of the printer ports and select **Accept**. The previous panel is displayed.

Byte Stream File from Keyed File

You can create a byte-stream file using a keyed file as input. The Keyed File Utility reads the keyed file sequentially and writes the records to a new byte-stream file.

To create a new byte-stream file from an existing keyed file:

1. Select **Byte Stream File from Keyed File** from the Create options pull-down.
2. Type the name of the existing keyed file or select it from the list.
3. Type the name for the new byte-stream file that will be created, or select a name from the list. If the file already exists, a panel is displayed that prompts you to verify that the file should be overwritten. If you do not want to overwrite the file, select **No** and you will be returned to the Create Byte Stream File from Keyed File panel to enter a different file name.
4. Put a check in the **Erase Input File** check box to erase the keyed file after the byte-stream file is created. Do not put a check in the check box to save the keyed file.
5. Select **Accept** when all information is complete to create the byte stream file. The Create Byte Stream File Status panel displays the following information:
 - Elapsed time
 - Input file
 - Output file
 - Percent complete

The percent complete status bar displays the progress of your request. You can cancel the request at any time by selecting **Cancel**. If the request fails, an error message is displayed and then the main panel is displayed.

6. Select **Return** when you have finished.

Empty Keyed File

You can create an empty keyed file to which applications can add records as needed.

To create an empty keyed file:

1. Select **Empty Keyed File** from the Create options pull-down.
2. Type the name of the keyed file you are creating or select it from the list. If the file already exists, a panel is displayed that prompts you to verify that the file should be overwritten. If you do not want to overwrite the file, select **No** and you will be returned to the Create Empty Keyed File panel to enter a different file name.
3. Type the record length for the file (in the range 1 to 4,092).
4. Type the key length for the file (must be less than or equal to the record length).
5. Type the number of records for the file.
6. Select the hashing method to use for the file. See “[Hashing Algorithms](#)” on page 68 for more information about hashing methods.

Based on your input, the system calculates defaults for the following values and uses them to create the new keyed file:

- Block size
- Randomizing divisor
- Chaining threshold

You can select **Change Calculated Defaults** to view or change the default values. The window that is displayed lists the values that have been calculated from the values you indicated on the Create Empty Keyed File panel.

You can change the calculated values or you can exit from the panel and return to the Create Empty Keyed File panel.

If you change the calculated values, the values on the Create Empty Keyed File panel do not change. Similarly, if you have changed the calculated default values, and then change values on the Create Empty Keyed File panel, the default values you changed are still in effect.

If you have changed the calculated values and want to recalculate the values based on the values on the Create Empty Keyed File panel, select **Use Dynamic Defaults**.

7. Select **Accept** when all information is complete to create the keyed file. The Create Keyed File Status panel displays the following information:
 - Elapsed time
 - Output file
 - Status messages

Messages are displayed to tell you the status of your request. You can cancel the request at any time by selecting **Cancel**.

Report

Select Report from the main panel. The following options are displayed:

- Keyed File Statistics
- Chaining Statistics for Byte Stream File

Keyed File Statistics

This function displays statistics on each keyed file that is created. The statistics include the number of:

- Reads, writes, deletes, opens, and closes
- Blocks
- Chains

You can examine these statistics and clear them.

To create a report on keyed file statistics:

1. Select **Keyed File Statistics** from the Report options pull-down.
2. Type the name of the keyed file or select it from the list.
3. Select **Reset Statistics** to clear the collected statistics.
4. Select **Accept** to view the performance statistics. Messages are displayed to tell you the status of your request. You can cancel the request at any time by selecting **Cancel**.
5. You can browse the status messages and select **Return** when you have finished, or you can save or print the status messages.

Select **Save Status** to save the messages. The Save Request Status panel is displayed and you are prompted to select the output destination as either a file or a printer port:

- To save the status messages to a file, select **File** as the output destination, type the output file name or select it from the list, and select **Accept**. The previous panel is displayed.
- To print the status messages, select one of the displayed printer ports and select **Accept**. The previous panel is displayed.

Chaining Statistics for Byte Stream File

You can test the efficiency of randomizing divisors and hashing algorithms before you create a keyed file. The Keyed File Utility scans the byte-stream file to report the percentage of chained records and the distribution. See “[Record Chaining](#)” on page 72 for more information about record chaining.

To create a report on the chaining statistics of a byte-stream file:

1. Select **Chaining Statistics for Byte Stream File** from the Report options pull-down.
2. Type the name of the byte-stream file or select it from the list.
3. Type the record length for the file (in the range 1 to 4,092).
4. Type the key length for the file (must be less than or equal to the record length).
5. Type the percentage of extra space to be left available in the new file.
6. Select the hashing method to use for the file. See “[Hashing Algorithms](#)” on page 68 for more information about hashing methods.
7. Select **Accept**.

Based on your input, the system calculates defaults for the following values:

- Block size
- Randomizing divisor

You can select **Change Calculated Defaults** to view or change the default values. The window that is displayed lists the values that have been calculated from the values you indicated at the Report Chaining Statistics for Byte Stream File panel.

You can change the calculated values or you can exit from the panel and return to the Report Chaining Statistics for Byte Stream File panel.

If you change the calculated values, the values on the Report Chaining Statistics for Byte Stream File panel do not change. Similarly, if you have changed the calculated default values, and then change values on the Report Chaining Statistics for Byte Stream File panel, the default values you changed are still in effect.

If you have changed the calculated values and want to recalculate the values based on the values on the Report Chaining Statistics for Byte Stream File panel, select **Use Dynamic Defaults**.

8. Select **Accept** to view the statistics.
Messages are displayed to tell you the status of your request. You can cancel the request at any time by selecting **Cancel**.
9. You can browse the status messages and select **Return** when you have finished, or you can save or print the status messages.

Select **Save Status** to save the messages. The Save Request Status panel is displayed and you are prompted to select the output destination as either a file or a printer port:

- To save the status messages to a file, select **File** as the output destination, type the output file name or select it from the list, and select **Accept**. The previous panel is displayed.
- To print the status messages, select one of the printer ports displayed and select **Accept**. The previous panel is displayed.

Verify

Select Verify from the main panel. The following option is displayed:

- Keyed File Integrity

Keyed File Integrity

To verify the integrity of a keyed file:

1. Select **Keyed File Integrity** from the Verify options pull-down.
2. Type the name of the keyed file or select it from the list.
3. Type the maximum number of errors you want to report.
4. Select **Accept**. The Keyed File Integrity Verification Status panel displays the following information:
 - Elapsed time
 - Input file
 - Status messages
 - Percent complete

Messages are displayed to tell you the status of your request. You can cancel the request at any time by selecting **Cancel**.

5. You can browse the status messages and select **Return** when you have finished, or you can save or print the status messages.

Select **Save Status** to save the messages. The Save Request Status panel is displayed and you are prompted to select the output destination as either a file or a printer port:

- To save the status messages to a file, select File as the output destination, type the output file name or select it from the list, and select Accept. The previous panel is displayed.
- To print the status messages, select one of the printer ports displayed and select Accept. The previous panel is displayed.

Using the Keyed File Utility From the Command Line

All of the functions for the Keyed File Utility can also be initiated from a command line.

Functions that can be performed are:

- Create a new keyed file from an existing byte-stream file
- Create a new byte-stream file from an existing keyed file
- Create an empty keyed file
- Report keyed-file performance statistics
- Report the chaining statistics for a byte-stream file
- Restart from a checkpoint
- Cancel (erase) a checkpoint
- Verify keyed-file integrity

These functions can be performed from any workstation in the DDS/CSF system. DDS must be running on the local workstation for a command to be successful.

See “[Starting the Keyed File Utility](#)” on page 57 for detailed descriptions of each of the Keyed File Utility functions.

See “[FDSKFU](#)” on page 65 for information about using the command-line interface to the Keyed File Utility.

FDSKFU

Purpose

Provides a command-line interface to Keyed File Utility functions.

You can obtain help for each of the functions provided by **FDSKFU** by typing **FDSKFU ?** and pressing **Enter**.

Syntax

The syntax for this command varies according to the function you want to use. The *drive* and *path* are the drive and path where DDS/CSF is installed.

Note: In the following list of commands, brackets ([]) are used to denote optional keywords.

Creating a keyed file from a byte stream file:

drive path FDSKFU 1 *ip od inname outname bs rs ks nr rd ct [ha]*

Creating a byte stream file from a keyed file:

drive path FDSKFU 2 *ip od inname outname*

Reporting keyed file statistics:

drive path FDSKFU 3 *ip inname*

Creating an empty keyed file:

drive path FDSKFU 4 *od outname bs rs ks nr rd ct [ha]*

Reporting chaining statistics from a byte stream file:

drive path FDSKFU 5 *inname bs rs ks nr rd [ha]*

Restarting from a checkpoint:

drive path FDSKFU 6

Restarting from a checkpoint if the checkpoint exists for a keyed file:

drive path FDSKFU 7 *ip od inname outname bs rs ks nr rd ct [ha]*

If the checkpoint exists for a different keyed file, this command will delete the checkpoint record and create the specified keyed file from a byte-stream file. There are three unique conditions:

1. No checkpoint. The command creates a keyed file from a byte-stream file.
2. Checkpoint matches specified keyed file. The command restarts from a checkpoint.

3. Checkpoint does not match specified keyed file. The command cancels the checkpoint, erases the checkpoint records, and creates a keyed file from a byte-stream file.

Canceling (erasing) a checkpoint:

drive path FDSKFU 8

Canceling the checkpoint and creating a keyed file:

drive path FDSKFU 9 *ip od inname outname bs rs ks nr rd ct [ha]*

Verifying the integrity of a keyed file:

drive path FDSKFU 10 *inname [me]*

Parameters

ip Indicates disposition of input file after processing is finished. Valid values are:

0 Do not erase the input file.

1 Erase the input file.

If you are reporting keyed-file statistics, *ip* indicates the disposition of the keyed file statistics. Valid values are:

0 Do not reset statistics.

1 Reset (clear) statistics.

od Indicates disposition of existing version of the output file. Valid values are:

0 Do not erase the existing version of the output file.

1 Erase the existing version of the output file.

inname Indicates the name of the input file.

outname Indicates the name of the output file.

bs Indicates the block size of the output keyed file. Valid values for this parameter are in the range from 512 to 4,096, in multiples of 512.

rs Indicates the record length of the output file. Valid values for this parameter are in the range from 1 to block size minus 4.

ks Indicates the length of key. This value must be less than or equal to the record length.

nr Indicates the number of records to be allocated on disk. This value should be the maximum number of records anticipated plus 25%.

- rd* Indicates the randomizing divisor. This value must be less than or equal to the number of blocks in the file. If you specify 0 (zero), the most effective randomizing divisor is calculated by the Keyed File Utility.
- ct* Indicates the chaining threshold. This value must be less than the number of blocks in the file.
A value of 0 indicates that there will be no probes related to record chaining for the specified file. See “[Record Chaining](#)” on page 65 for more information about chaining thresholds and record chaining.
- ha* Indicates the hashing algorithm to be used. This value is optional. See “[Hashing Algorithms](#)” on page 68 for detailed information about the hashing algorithms. Valid values for this parameter are: 0 IBM folding algorithm (4680 Compatibility only) 1 XOR rotation algorithm (4680 Compatibility only) 2 Polynomial algorithm (default)
- me* Indicates the maximum number of errors that the file validation function should report. This value is optional. The default value for this parameter is 10.

Remarks

DDS must be running on the local workstation for this command to be successful.

The selection of an efficient randomizing divisor is important in minimizing the number of chained records in a keyed file. Chained records require more disk read operations for retrieval than do home records. Therefore, minimizing chained records maximizes performance when accessing the file by key.

Before selecting a randomizing divisor, calculate the number of blocks in the file as follows:

$$\text{Records per block} = (\text{block size} - 4) / \text{record size} \quad \text{Total blocks} = \text{total records} / \text{records per block}$$

The randomizing divisor must be less than or equal to the total blocks in the file. The chaining threshold must be less than the total blocks. In general, prime numbers make the most efficient randomizing divisors. You might choose the highest prime number that is less than or equal to the number of blocks in the keyed file.

See “[Record Chaining](#)” on page 65 for more information about record chaining.

Examples

1. Creating a keyed file from a byte-stream file:
FDSKFU100 input.fil prices.key 512 50 7 50000 0 4
2. Creating a byte-stream file from a keyed file:
FDSKFU200 prices.key output.fil

3. Reporting keyed file statistics:
FDSKFU 3 0 prices.key
4. Creating an empty keyed file:
FDSKFU 4 0 empty.key 512 50 7 50000 0 4
5. Reporting chaining statistics from a byte-stream file:
FDSKFU 5 input.fil 512 50 7 50000 0
6. Restarting from a checkpoint:
FDSKFU 6
7. Restarting from a checkpoint if the checkpoint exists for a keyed file:
FDSKFU700 input.fil prices.key 512 50 7 50000 0 4
8. Canceling (erasing) a checkpoint:
FDSKFU 8
9. Canceling the checkpoint and create a keyed file:
FDSKFU900 input.fil prices.key 512 50 7 50000 0 4
10. Verifying integrity of a keyed file:
FDSKFU 10 prices.key

Return Codes

- | | |
|-----------|----------------------------|
| 0 | Operation was successful |
| 1 | DDS is not running |
| 2 | Incorrect parameter |
| 3 | Command canceled by user |
| 4 | Internal error |
| 5 | Out of memory |
| 6 | Program is already started |
| 99 | Command canceled by user |

Hashing Algorithms

Hashing algorithms are key-transformation algorithms that randomize the distribution of keyed records in a keyed file. Randomizing transforms record keys into expected block addresses relative to the start of the keyed file.

There are two steps to hashing algorithms:

1. Key folding
The bytes of the key are logically accumulated into 2 accumulation bytes. These bytes are initialized to 0 (zero). Alternate key bytes are folded into accumulators with the Exclusive OR (XOR), starting with the low-order key byte into the low-order accumulator and the next byte of the key into the high-order accumulator. This process continues, working through the key from low-order to high-order (right to left), until the key is entirely processed. The result is a 2-byte value derived from the entire key.
2. Randomizing

A randomizing divisor divides the folded key. The divisor must be less than or equal to the number of blocks allocated for the keyed file. The remainder from the division is the expected, relative block number. The choice of the randomizing divisor and record keys affects the distribution of records in the keyed file. You must choose these variables so that the keyed file contains a uniform distribution of records. Although no records randomize greater than the randomizing divisor, these blocks are used for overflow records if any additional blocks are needed and available.

Randomization is performed by dividing the 2-byte folded key by the randomizing divisor. The resulting remainder is the relative block number in the keyed file that contains the keyed logical record or an overflow chain pointer to the block containing the record. If the remainder is equal to 0, then the relative block number is equal to the randomizing divisor.

If you select a randomizing divisor rather than having the Keyed File Utility create one, remember that the selection of an efficient randomizing divisor is important in minimizing the number of chained records in a keyed file. Chained records require more disk read operations for retrieval than do home records. Therefore, minimizing chained records maximizes performance when accessing the file by key.

Before selecting a randomizing divisor, calculate the number of blocks in the file as follows:

Records per block = (block size -4)/ record size

Total blocks = total records / records per block

The randomizing divisor must be less than or equal to the total blocks in the file. In general, prime numbers make the most efficient randomizing divisors. You might choose the highest prime number that is less than or equal to the number of blocks in the keyed file.

See “[Record Chaining](#)” on page 72 for more information about chained records.

There are three hashing algorithms available with the Keyed File Utility:

- Polynomial Hashing Algorithm
- IBM Folding Algorithm
- XOR Rotation Hashing Algorithm

Note: The IBM Folding Algorithm and XOR Rotation Hashing Algorithm are provided only for 4680/4690 compatibility. See the *IBM 4690 OS Programming Guide* for more information. The polynomial hashing algorithm should be used for all new files.

Polynomial Hashing Algorithm

When the polynomial hashing algorithm is used on large keyed files, it results in fewer chained records and improved performance in file build and file access. (See “[Record Chaining](#)” on page 72 for more information about chained records.)

Note: The IBM Folding Algorithm and XOR Rotation Hashing Algorithm are provided only for 4680/4690 compatibility. See the *IBM 4690 OS Programming Guide* for more information. The polynomial hashing algorithm should be used for all new files.

The polynomial hashing algorithm improves record distribution when keys contain repeated numeric patterns. A key that has ASCII characters is an example of a key with repeated numeric patterns.

The polynomial hashing algorithm uses the same technique that generates the frame-check-sequence (FCS) field for the synchronous data link control (SDLC) transmission frame. The polynomial hashing algorithm is for users who generate the FCS without using the Keyed File Utility. Refer to *IBM Synchronous Data Link Control Concepts* for more information.

The algorithm performs the following steps:

1. Generates a 16-bit FCS number from the key of a record.
2. Reverses the key byte by byte.
3. Generates another 16-bit FCS number from the reversed key.
4. Concatenates the 16-bit numbers from step 1 and step 3 to make a 32-bit number. The high-order 16 bits are from step 3 and the low-order 16 bits are from step 1.
5. Turns off the high-order bit of the 32-bit number from step 4.
6. Divides this 32-bit number by the randomizing divisor. The remainder becomes the relative block of the record.

Example of Polynomial Hashing Algorithm

The following example illustrates the polynomial hashing algorithm. You can use the following method to generate a keyed file without using the Keyed File Utility.

Assume a 4-byte record key of 10530237 and a randomizing divisor of 49997.

Key	16-Bit FCS Number	Hexadecimal Equivalent	Bit Notation
10530237	9588	2574	0010 0101 0111 0100
Reversed Key 37025310	16-Bit FCS Number 48043	Hexadecimal Equivalent BBAB	Bit Notation 1011 1011 1010 1011

Concatenation of these two numbers equals:

1011 1011 1010 1011 0010 0101 0111 0100

When you turn off the high-order bit, the number equals:

0011 1011 1010 1011 0010 0101 0111 0100

The hexadecimal representation of the result equals:
3BAB2574

The decimal value equals:
1,001,071,988

Divide the decimal value (1 001 071 988) by the randomizing divisor (49997) giving:
20022 Remainder = 32054

The home block of the key (10530237) equals:
Remainder = 32054 Hexadecimal Value = 7D36

Sample C-Language Program

The following sample program is written in C language. It is used to generate a 16-bit FCS number.

```
unsigned short crcgen (data, length)
char *data; /* address of the key */
int length; /* key length */
{
    unsigned char crcbyte1, crcbyte2, r1;
    char *ptr;
    int i;
    union byte_to_word
    {
        char bytes[2];
        unsigned short ret;
    } return_val;

    ptr = data;
    crcbyte1 = 0xFF
    crcbyte2 = 0xFF

    for (i=0; i<length; i++)
    {
        r1 = *ptr++ | crcbyte2;
        r1 = (r1 << 4) | r1;
        crcbyte2 = (r1 << 4) | (r1 >> 4); crcbyte2 = (crcbyte2 & 0x0F ) | crcbyte1;
        crcbyte1 = r1; r1 = (r1 << 3) | (r1 >> 5); crcbyte2 = crcbyte2 | (r1 & 0xF8);
        crcbyte1 = crcbyte1 | (r1 & 0x07);
    }
    return_val.bytes[0] = ~crcbyte2;
    return_val.bytes[1] = ~crcbyte1;
    return(return_val.ret);
}
```

The following list shows the output from the sample program when used with the input data shown. The I/O is 2 bytes and in hexadecimal form.

Input	Output
1234	DEC1
1111	8206
4143	2066

Record Chaining

Each block has two chain pointers, and each chain pointer is 2 bytes long. One pointer points forward and the other points backward. These pointers are used to chain together records that overflow from one block into another. By chaining records using pointers, it is known to which home block the data is related.

The block to which the overflow data is related is called a *home block*. If you try to add a record to a home block that is already full of records, a check is made to see whether an overflow chain exists for that block. If one does, a check is made for available space. If available space is found, the record is added to the block with the available space. If there is no available space in that overflow chain, another block is located without overflow records, and the record is added to the end of that block's overflow chain.

If no overflow chain exists for the original home block, then one is created. The keyed file is scanned for a block that has sufficient room for the record and that does not already contain other overflow records. The record is added to this block and this block is added to the record's home block overflow chain.

Note: Overflow chains cannot contain overflow records from more than one home block. When creating a keyed file, the system sets a value called the *chaining threshold*.

Each time the number of chains traversed to write a record exceeds the chaining threshold, a message is logged in the Windows event logs. This message indicates that either the keyed data storage is getting too full or that the file is poorly randomized.

Figure 1 through Figure 3 illustrate examples of space management with and without overflow chains.

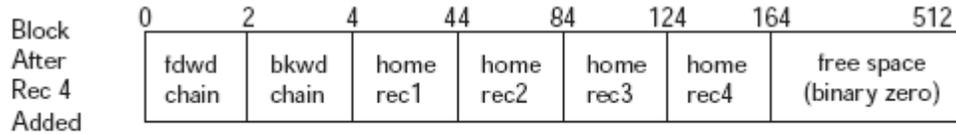
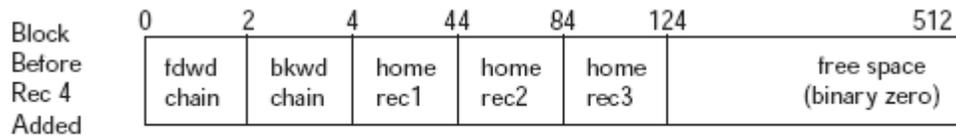


Figure 1. Home-Record Add without Overflow Present

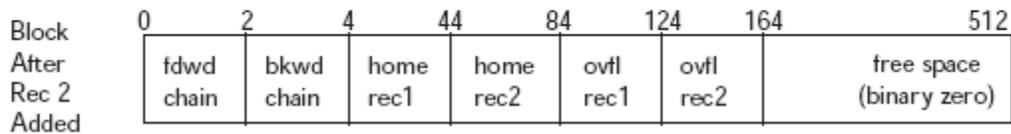
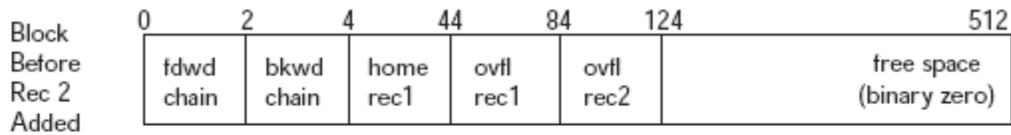
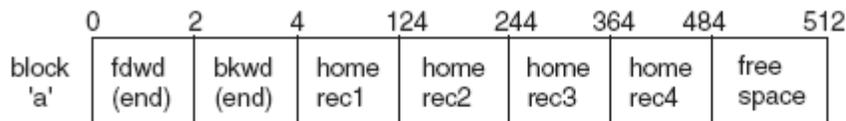


Figure 2. Home-Record Add with Overflow Present

Keyed file before record five is added to block 'a'.



Keyed file after the overflow block is allocated and record five is added. Record five becomes the first overflow chain for block 'a'.

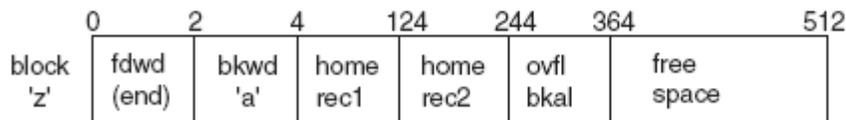
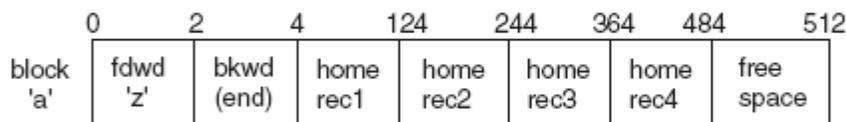


Figure 3. Example of Adding a Record to a Full Home Block Creating Overflow Chain

Chapter 7. DDS/CSF Provides MCF Function

DDS provides two optional features for customers who are currently using the 4690 Operating System (OS) and want to preserve the investment in 4690 applications, terminal hardware, and terminal wiring while moving to an open platform for the back room:

- **DDS 4690 Controller Services Feature (CSF)**
CSF provides support for running 4690 OS controller functions and applications on Windows. It provides many of the functions of the 4690 OS.
- **DDS 4690 Multiple Controller Feature (MCF)**
MCF provides support for using multiple controllers. It provides many of the functions supported by the 4690 Multiple Controller Feature.

Supported Functions and Capabilities

DDS/CSF is not designed to be full-function replacements for the 4690 OS and the 4690 Multiple Controller Feature. Many 4690 functions can be performed with your Windows operating system and need not be reproduced in CSF and MCF. The following sections describe the functions and capabilities of CSF and MCF.

CSF Functions

Controller application execution

No 4680 BASIC application changes or recompiling or relinking are required. The same binary format used on the 4690 OS can run unmodified on Windows using this feature.

Varied terminal attachment

Terminals can be attached via store loop, token ring, Ethernet or wireless communications (Symbol or Telxon). Terminals can be loaded and dumped, and terminal-to-controller communication is supported.

Terminal backup

Terminal backup can be performed on store loop, token ring, or Ethernet when two workstations running 4690 CSF are attached to the terminals.

Keyboard control

System control functions can be accessed by the ALT+F12 keys.

Host communications

Host communications use TCP/IP.

4690-style logon and navigation

You can perform normal 4690 logon sequences and screen navigation.

Background application support

Applications can run in the background. They can be started when the DDS 4690 CSF is started, by other applications, or by the correct key sequences on the background control panel.

Multitasking

Multiple 4690 windows can run concurrently, each with the capability of running applications concurrently.

Standard 4690 applications

You can use all of the following standard 4690 system applications:

- Change Configuration Report
- Configuration Input State Table Build Utility
- Remote Command Processor
- System Log Formatter
- Compress/Decompress
- Audible Alarm
- Terminal Dump Formatter
- Dump Analyzer for Terminal Dumps

4690 event log

The 4690 event log is supported.

4690 Device Channel Trace

You can run the 4690 Device Channel Trace on the terminals as requested by DDS 4690 CSF. The Device Channel Trace Report is also available.

4690 Controller RAM disk

The 4690 Controller RAM disk function is available. See “[Helpful Tips and Limitations](#)” on page 90 for more information.

DDS/CSF Supports MCF Functions

MCF provides the following functions and capabilities:

File distribution

The DDS Data Distribution component provides for file distribution that is functionally similar to that provided by the 4690 OS and the 4690 Multiple Controller Feature.

DDS also provides file attributes that determine the target and frequency of the distribution. In the DDS system, the target of the distribution can be local, mirrored, or broadcast. Distribution frequency can be either distribute on update (DOU) or distribute on close (DOC).

Node roles

In a typical 4690 configuration, there will be at least two controllers: one serves the role of a master and file server, and the other serves the role of alternate master and alternate file server. Other machines in the system typically serve the role of subordinate.

DDS provides support for node roles that are functionally similar to the roles provided by 4690.

Note: DDS MCF does not provide subordinate support.

4690 Node Roles:	DDS Node Roles:
Master and File Server	Primary Distributor
Alternate Master and alternate File Server	Backup Distributor
Subordinate	Not supported

4690 Functions Not Supported or Changed in DDS/CSF

This section describes 4690 functions that are not supported by DDS/CSF, as well as those that differ from 4690 OS.

- On the File Utilities window:
 - **Keyed File Utilities:** This function is provided by DDS.
 - **Display/Alter File Data:** This utility requires the 4690 FAT system, which is not possible with CSF.
 - **Distributed File Utilities:** This function is provided by DDS.
 - **Streaming Tape Drive Utilities and Optical Tape Drive Utility:** These utilities are used for 4690 OS backup and are not supported. There are utilities available in Windows that provide this function.
- On the Installation and Update Aids window:
 - **Apply Software Maintenance and Build Software Maintenance Control File:** These functions must be handled by facilities used to support the rest of the Windows operating system.
- On the Problem Analysis Data Collection window:
 - **Device Channel Trace:** This is the only trace supported by CSF.
 - **Start Performance Data Collection and Stop Performance Data Collection:** These features are not supported. There are performance tracking tools available on Windows operating systems that provide this function.
 - **Start Trace Data Collection and Stop Trace Data Collection:** These options can be used only for Device Channel Trace.
- On the Problem Analysis Reports window:
 - **Format Performance Data:** No performance data is collected.

- Create Problem Analysis Diskette: Use the DDS Problem Analysis Utility.
- **Store Controller Services:** CSF does not support the programmable power functions for controllers. They include powering off a remote controller, disabling controller programmable power, enabling controller programmable power, and powering off a local controller.
- **SNA Communications:** This function is not supported under CSF.
- **ADCS Support:** This function is not supported under CSF.
- **Artic Card:** This feature is not supported under CSF.
- **ASM using the JAVA interface:** This feature is not supported under CSF.
- **4690 Application Debugger:** CSF cannot support the debugger when used to debug a controller application because 4690 OS uses system control functions that no version of the Windows Operating System has. The 4690 Application Debugger can be used, however, to debug terminal application programs.
- **System keys:** System keys are presented by pressing Alt-F12. Alt-SysRq does not work in CSF. Selection **t** (Switch to Terminal Mode) cannot be used because controller/terminal support is not available in CSF. However, it is possible to run CSF and Terminal Services in the same controller. Switching back and forth requires Windows-defined keystrokes.
- **Multiple Controller System Function:** The Multiple Controller System Function option is not supported. Use the DDS Node Control Utility to perform these functions.
- **Diskette drive support:** CSF does not support the movement of files to and from diskette. The use of **A:** is not supported. Use a Windows operating system command prompt for diskette-drive activity.
- **Dump Controller Storage:** This option works differently in CSF than it does in 4690 OS. Instead of creating the controller dump file called ADXCSLCF.DAT in the root directory, CSF creates a SYSTEM.PAN file in the log subdirectory \$FDSDIR\LOG.
- **File compression:** File compression using a list of files operates correctly on CSF. Note that the list of files to be compressed can be created on a native 4690 OS system or directly on a Windows system, but you must still format the file containing the list as described in 4690 OS documentation.
- **File deletion:** If an application running in a CSF environment attempts to delete an open file, the application will receive a return code indicating that the delete was unsuccessful. This also applies to Windows operating system applications using the 4690 C API or Java™ applications that use the 4690 Java API. When running under the 4690 native environment, the application receives a return code indicating that the delete was successful; however, the file remains on the drive until the file is closed.
- **4690 Communication and Systems Management Facility:** This facility is not operational in CSF. Alerts sent to NetView® when certain 4690 events are logged will not flow to the host when these events are logged by CSF. This statement applies to both system alerts and user application alerts.
- **3270 emulation:** CSF does not support 3270 emulation from the Windows controller. Use Windows-based applications to provide the same function.
- **Remote Operator functions:** CSF does not support the use of 4690 OS Remote Operator functions.

- **-p parameter for 4690 COMMAND command:** CSF does not support use of the -p parameter on the COMMAND command. If this parameter is used, the command will fail. All other functions of this command function as described in the *IBM 4690 Operating System: User's guide*.
- **IPL command processor:** CSF does not support the 4690 OS IPL command processor.
- **OEM adapters:** CSF does not support unique OEM adapter cards.
- **ADX_CLOCK_REC:** The application programming interface allows a flag setting of 1, which requests an exclusive lock. Do not use 3, which requests a shared write lock. **If option 3 is used CSF will generate a dump if the Application Dump flag is set to Y in the 4690 OS controller configuration (under Controller Characteristics).** A flag setting of 2, which in 4690 applications is *exclusive write lock* is implemented in CSF as exclusive lock (flag setting 1).
Also, you cannot unlock a portion of a locked area with CSF, as you can with 4690. You must unlock the entire area.
- **Java POSFile lock method:** Do not use the **LOCK_SHARED** value of the lockType parameter because this will cause CSF to generate a dump if the Application Dump flag is set to Y in the 4690 OS controller configuration (under Controller Characteristics). You can use the other values of the lockType parameter.
- **Configuration of Java Terminal Off Line (TOF):** This configuration is not supported under DDS/CSF, so all configuration and setup of Java terminal offline must be done on your 4690 OS library controller before you copy the 4690 OS files to your Windows controller.
- **32 bit 4690 OS applications:** CSF does not support these applications.

Windows changed and removed:

- The 4690 System Control Functions window, Multiple Controller Functions (window CSO3S064) has been removed. In a 4690 configuration, this window allows you to control the activation and deactivation of the 4690 master and 4690 file server. DDS provides the Node Control Utility and APIs to control the activation and deactivation of the primary distributor.
- The Distributed File Utilities (ADXCSU0L) listed on the File Utilities window has been removed. DDS provides the Data Distribution Utility and APIs to control data distribution. The Data Distribution Utility also controls the exception log for data distribution.
- 4690 file distribution functions, APIs, and attributes are not used by MCF. All distribution functions are controlled through the DDS Node Control Utility, Data Distribution Utility, and Data Distribution APIs.

Note: 4690 Multiple Controller Feature functions, such as running the Distributed File Utility and applications accessing byte-stream files on other nodes, are supported by DDS/CSF.

CSF Functions in Detail

TCP/IP Support

4690 OS TCP/IP support is included in CSF. The 4690 OS TCP/IP support is documented in the IBM Transmission Control Protocol/Internet Protocol for 4690 Applications Interface Document. The TCP/IP support in CSF is based on the TCP/IP stack provided by Windows. It is the Windows TCP/IP stack that performs the actual communication with the network, and, therefore, it must be installed and configured correctly to allow the 4690 OS TCP/IP support in CSF to function properly.

- The only supported interfaces and APIs are:
 - Socket Programming Interface
 - File Transfer Protocol (FTP) Interface

- The following functions, commands, and interfaces are not supported:
 - Simple Network Management Protocol (SNMP) Agent.
 - Interned superserver (INETD)
 - BOOTP client and server
 - SNMP Distributed Program Interface (DPI)
 - Telnet client and server, including the Telnet command
 - LPR client
 - File transfer protocol (FTP) client and server
 - Trivial file transfer protocol (TFTP) commands
 - Network File System (NFS) server
 - Network File System (NFS) client in the controller
 - Virtual File System (VFS) server and client
 - Portmapper
 - Netstat
 - Finger
 - Rexec client
 - VT100 client
 - Sun Remote Procedure Call (RPC)

Windows provides applications and servers with its TCP/IP stack and extensions (such as Peer Web services) to provide the functions listed above.

Other limitations and restrictions imposed by Windows TCP/IP API are:

accept

For sockaddr and addrlen, the address must be valid for the connecting entity or NULL. The address length must be at least the size of the space pointed to by the address, or NULL.

getsockopt

The following settings for the optname are not supported:
SO_RCVLOWAT, SO_RCVTIMEO, SO_SNDLOWAT,

SO_SNDTIME, SO_USELOOPBACK. In addition, the following level and optname combinations are not supported (level/optname): stream/SO_BROADCAST, datagram/SO_KEEPALIVE, datagram/SO_OOBINLINE, raw/SO_KEEPALIVE, raw/SO_OOBINLINE.

ioctl

Only the following subset of Berkeley Sockets ioctl commands is supported: FIONBIO, FIONREAD, and SIOCATMARK.

setsockopt

The following settings are not supported for optname: SO_RCVLOWAT, SO_RCVTIMEO, SO_SNDLOWAT, SO_SNDTIMEO, SO_USELOOPBACK. In addition, the following combinations of level and optname are not supported (level/optname): stream/SO_BROADCAST, datagram/SO_KEEPALIVE, datagram/SO_OOBINLINE, raw/SO_KEEPALIVE, raw/SO_OOBINLINE.

socket

Windows does not support all the same protocols as 4690 OS. For example, for raw sockets, the protocol of 256 is not valid for Windows. The user program must check the return code from the socket call to ensure that it is a non-negative response. A negative response indicates an error in attempting to create the socket with the specified parameters.

putunique (in the FTP APIs)

The putunique FTP API that uses the FTP subfunctions sunique is not supported by the Windows FTP server. Sunique is used to toggle on and off the state indicating whether FTP needs a unique name to write the files

However, putunique may operate on non-Windows FTP servers, depending on the capability of the FTP server on the server machine.

sendto

Windows TCP/IP does not support the flag values of blank and MSG_PEEK for the UDP (datagram sockets). Also, the packet size sent on a datagram socket cannot exceed the maximum message size.

4690 OS TCP/IP supports a number of TCP/IP data files that have Windows TCP/IP counterparts. These files will continue to exist with CSF in order to better support current 4690 OS TCP/IP applications. These files are:

- adxhsipf.dat (protocol file)
- adxhsisf.dat (services file)
- adxhsinf.dat (networks file)
- adxhsirf.dat (resolve file)
- adxhsihf.dat (hosts file)
-

Windows does not use or recognize these files.

Additional TCP/IP considerations:

Installation and configuration:

Installation and configuration of the 4690 OS TCP/IP support should be performed on your native 4690 OS library machine before transferring the 4690 OS files to your CSF Windows controller. (See later sections in this chapter for more information about transferring these files.) This includes setting up the configuration batch file (ADXIPxxZ.BAT, where xx is your controller ID) and any of the data files, such as the hosts file.

The configuration batch file is very important. It triggers the loading of the TCP/IP driver in CSF, and sets up the driver properly. Note that the configuration of background 4690 OS applications should not include any background-executed TCP/IP applications, which CSF does not support. If you already have background 4690 OS applications, they should be examined and altered as needed to prevent them from attempting to execute unsupported TCP/IP applications or services.

ROUTE function

The ROUTE (adxhsi41) function (which is typically executed via the ADXIPxxZ.BAT file when the 4690 OS TCP/IP driver is loaded) is not required in the CSF environment. All sockets-related activities are handled by the Windows TCP/IP interfaces, which use the Windows TCP/IP settings for their actions.

Windows also provides its own ROUTE application which can be used to manipulate the routing-table entries. The presence of calls to the ROUTE function in the ADXIPxxZ.BAT file will not affect processing in the CSF environment.

IFCONFIG function

The IFCONFIG (adxhsi31) function should only be used with the ADXIPxxZ.BAT file in order to establish the hostid within the CSF TCP/IP driver. This setting is needed so the gethostid API call has a value to return. This value should match the Windows TCP/IP setting for the IP address (as set up in Control Panel/Network/Protocols/TCP/IP/IP address). The Control Panel in Windows should be used to change the IP address if required.

Stack socksification

On your Windows controller, running CSF with 4690 OS TCP/IP, socksification or automatic socks software must be running in order to use the FTP APIs to communicate outside a firewall. Socksification or automatic socks software improve the performance of Windows TCP/IP and allow secure communications through firewalls. The Windows FTP client has the same requirement for communication outside the firewall.

Configuration of socksification or automatic socks software may require some changes if you are experiencing unexpected results when running your 4690 OS TCP/IP application.

For example, an application may find that it cannot connect any UDP/datagram sockets to a particular server. If the socksification software has a redirection entry in it that identifies that both TCP and UDP requests will be routed through proxy socks server, try changing that single redirection request into two requests: one in which the TCP requests are routed

through the proxy socks server and one in which the UDP requests are routed directly to their destination server (no redirection).

Also note that any servers with which you directly communicate using the FTP APIs will use TCP and UDP connections. If your FTP API data requests (get, put, and so on) are failing to establish a data connection, you may need to modify your redirection requests for those servers.

Auxiliary Console Support

CSF provides support for 4690 OS auxiliary consoles. This support is limited to LAN-attached auxiliary consoles (via Telnet) and local auxiliary consoles (windows) on the Windows controller where CSF is running. All auxiliary consoles must be defined in the 4690 OS configuration as Telnet auxiliary consoles, even those that are local auxiliary consoles (windows) on your Windows controller. Up to eight auxiliary consoles may be defined.

LAN-attached auxiliary consoles attach to the Windows controllers running CSF via Telnet. A telnet server must be running on the Windows controllers. The auxiliary console support in CSF supports the TelnetServer 2000 from Pragma Systems, Inc. at <http://www.pragmasys.com>. The Pragma Systems, Inc. Telnet client is recommended as the telnet client used by remote Windows systems.

After you have attached to your Windows controller via Telnet, enter **fdsecnat -a** to bring up the CSF panels.

Note that the TelnetServer 2000 must be configured with the Graceful Termination option. This option sends CSF a termination string if the telnet session ends, allowing CSF to clean up its end of the Telnet session and prepare for other Telnet sessions. The graceful termination option for Telnet Server 2000 must be configured with the string **CTRL-ALT-F9,SLEEP2**, which is associated with the executable FDSECNAT.EXE.

If you are using all of your defined LAN-attached auxiliary consoles, and the Telnet session for one of them is broken, then it takes two hours before you can telnet in and establish another auxiliary console. This period is the default time requirement in Windows TCP/IP support before the port is freed.

To change this value, change the Windows registry. Add a value of **KeepAliveTime** to the following key:

\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters

The data for this value is in milliseconds, so a value of 300 000 is five minutes.

Note that this change may affect other applications in your Windows controller and that editing the Windows registry must be done very carefully.

For detailed information regarding the KeepAliveTime registry value and other registry entries affecting TCP/IP, refer to the Microsoft Knowledge Base at <http://www.microsoft.com>. Search on KeepAliveTime.

4690 Command Line Support

All 4690 commands that can be executed from the 4690 OS command line are supported with the following exceptions:

- adxcsj01
- adxnsxz1
- backup
- chkdsk
- diskcomp
- diskcopy d
- iskset
- format
- fset
- restore
- vol

The print command is supported but only the -d (or -D) parameter is allowed. Specification of any other parameter will cause the print command to fail.

CSF Toolkit

CSF provides a toolkit that allows users to develop native Windows applications that use the 4690 OS C language APIs. The toolkit consists of a set of header files (*.h) and lib files to be used in the development of Windows applications using the C language. The toolkit header files are in the *target_install_directory\include\fds* directory and the lib files are in *target_install_directory\lib*.

Windows Command-Line Kickoff of Background 4690

Applications 4690 CSF provides a command named **FDSCSH0L** to allow users to start 4690 OS background applications from the Windows command line. The syntax of **FDSCSH0L** is either: *FDSCSH0L applname parm1 parm2 parm3 ...* or *FDSCSH0L @applist*.

In the first format, *applname* is the name of the 4690 application to be started, including the complete 4690OS path name with any file extension. The *applname* is followed by any parameters that would normally be typed at the 4690 command prompt.

In the second format, *appllist*, preceded by the @ character, is the name of an ASCII text file containing the 4690 applications to be started. Each application listed in this file must be on a separate line as follows:

applname parm1 parm2 parm3

Both formats may contain 4690 or DDS logical names. DDS logical names must be enclosed in "<>" delimiters. DDS logical names may be nested. For example, a string that is resolved from a DDS logical name may itself contain another DDS logical name, as long as all the logical names can be eventually resolved.

Only one instance of **FDSCSH0L** may be running at any time.

FDSCSH0L waits indefinitely for completion of each 4690 application, in sequence. Upon completion of each 4690 application, **FDSCSH0L** displays the application return code in the Windows application event log. The return code from **FDSCSH0L** is zero when successful. The following nonzero return codes may be returned:

- 100** The length of the application name and parameters is too long.
- 200** The application list file could not be opened.
- 300** The log file could not be opened.
- 400** Another copy of **FDSCSH0L** is already running.
- 500** No parameters were specified for **FDSCSH0L**.
- 600** The ADXSERVE driver could not be opened.
- 700** An end of DDS logical name delimiter was found with no start delimiter.
- 800** A start of DDS logical name delimiter was found with no end delimiter.
- 900** A DDS logical name is too long.
- 1000** An error was encountered initializing the interface to DDS.
- 1100** An error was encountered retrieving the DDS configuration.
- 1200** An error was encountered trying to create a DDS queue.
- 1300** An error was encountered trying to resolve a DDS logical name.
- 1400** CSF is not up so **FDSCSH0L** ends without attempting anything.
- 1500** An error was encountered with the list file. **FDSCSH0L** ends.

Messages are written to the Windows NT, Windows 2000, Windows XP, or Windows Server 2003 application event log to note the success or failure of the startup requests. Also, if the request is to start 4690 system applications, such as ADXCSN0L.286 (4690 event log report), the 4690 application writes messages to the Remote Command Processor status file, ADX_SDT1:ADXCSHSF.DAT, indicating the success or failure of the 4690 application.

3270 Emulation Support

No support for 3270 emulation is provided by CSF.

Java Support

CSF provides support for Java controller and terminal applications if you are using 4690 OS V2 or higher.

Java applications on the controller require the use of a Windows Java virtual machine (JVM). This JVM is automatically installed when DDS/CSF is installed on the controller. Please note that, if you are upgrading from an older version of DDS/CSF, the Java configuration environment variable, JAVA4690_CLASSPATH, will still be in place. This variable will override other java classpath configurations that may be set up automatically. This value should be the same as that used by the automated installation. It is recommended that you remove this environment variable, so that no confusion exists.

If you are planning on using an existing 4690 OS Java controller application, you should be aware of potential changes to the application that may be needed in order to run the application on Windows with CSF. See [“Porting 4690 OS Java Controller Applications to CSF”](#) on page 86 for more details.

Java applications on the terminal use the 4690 JVM and can run unchanged with CSF.

4690 OS Java Native Interface (JNI) Support

CSF provides support for the 4690 OS Java Native Interface (JNI) included in 4690 OS V2 and later versions. This interface allows Java applications to access 4690 OS APIs and functions.

Further details about the 4690 OS JNI and how to write applications to use it can be found in the *4690 OS Version 3: Programming Guide* in the chapter titled “Designing Applications With Java.”

Remember the following if you plan to use the 4690 OS JNI:

1. CSF must be running in order for the 4690 OS JNI calls to be successful.
2. File specifications that include drive letters must use drive letters that are applicable to 4690 OS. For instance, to reference a file on a controlled drive which to Windows NT, 2000, XP, or Server 2003 is the E: drive but to 4690 OS is the C: drive (based on the DDS logical name adxcsfh0), you would use C: in the file specification.

ICESoft Browser

The ICESoft browser is supported by CSF on both the controller and the terminal.

When running the browser on the terminal, it is important that you have a correct terminal classpath. In addition, be aware that the browser is affected by any defined NFS mounts.

NFS mounts that are defined but are not accessible can cause delays when starting the browser.

Porting 4690 OS Java Controller Applications to CSF

If you plan to use an existing 4690 OS Java controller application with CSF you need to be aware of how the Java application references files. The application may use complete file specifications that include a drive letter. For 4690 OS this drive letter will be C: or D:.

However, with CSF on Windows, the 4690 OS drive that is the equivalent of the C: drive on 4690 OS may be the Windows E: drive. The Windows JVM you have chosen to use will not be aware of this difference. If you run the Java application unchanged, it will try to reference files on the Windows C: drive, which is incorrect in this case.

Therefore, you have two choices: modify the Java application to reference the correct drives or modify your Windows NT, 2000, XP, or Server 2003 installation. Modifying your Windows installation would involve installing Windows on a drive other than C: or D: and then using the Windows C: and D: drives for your controlled drives. This would allow you to run the Java application unchanged because the files it references would now be on the C: or D: drive in Windows.

You should also be alert to the 4690 OS Java controller application's use of 4690 OS logical names. Any Windows JVM will not understand 4690 OS logical names and therefore, you must modify the Java application if your Java application uses 4690 OS logical names.

For optimum portability, Java applications should identify files without drive letters and without 4690 OS logical names in the file specification.

Fonts may also be an important consideration when porting 4690 OS Java controller applications to CSF on Windows. Some 4690 OS fonts may be incompatible with Windows fonts. You will detect this when you run the Java application under CSF on Windows, and you will have to choose an appropriate Windows font that is supported by the JVM you have installed on Windows NT, 2000, XP, or Server 2003.

If you experience problems in running the controller JVM under 4690, look at the file, `c:\fds\log\javainfo.log`, to see the actual command issued by CSF to start the JVM including full classpath info. Be aware that, if you migrate DDS/CSF from a Version 2 level, the environment variable, `JAVA4690_CLASSPATH`, is defined. It takes precedence over classpaths that have been defined as part of the DDS/CSF automated configuration. You should delete this environment variable, although it should be consistent with the updated installation.

Generally, if you want to put some new java code into the classpath, the JVM will use the controller classpath configured in the 4690 console.

Accessing 4690 Resources From Windows NT, Windows 2000, Windows XP, or Windows Server 2003 Applications

With 4690 Controller Services Feature, you can write Windows applications that use 4690 C APIs. This capability allows your applications to use 4690 resources as needed.

Since your application is not started by 4690 CSF, your application must determine if CSF is started yet before calling any 4690 C APIs. A new C API, `ADX_CCSF_INIT`, is available exclusively for Windows applications to make this determination.

`ADX_CCSF_INIT` returns one of the following return codes:

- `CSF_READY_RC`. This return code indicates that CSF is running.
- `CSF_NOT_READY_RC`. This return code indicates that CSF is not running. You may end your application, continue to call the `ADX_CCSF_INIT` API until you receive a `CSF_READY_RC` return code, or continue running the application without calling any 4690 C APIs.

CSF normally stops applications using the 4690 interfaces when it ends. Since your application is a Windows application, you may want the application to continue running even after CSF ends. The `ADX_CCSF_DONTKILL` API instructs CSF to continue running your application.

The `ADX_CCSF_DONTKILL` API must be called before any other 4690 C APIs are called, including the `ADX_CCSF_INIT` API. `ADX_CCSF_DONTKILL` returns one of the following return codes:

- `CSF_MUST_DONTKILL_FIRST_RC`. This return code indicates that you did not call this API first.
- `CSF_OK_RC`. This is a successful return code.

When using `ADX_CCSF_DONTKILL`, other 4690 C API calls may return codes indicating CSF is going to end. The return code values are:

- `CSF_SHUTDOWN_RC`. This return code indicates that CSF has shutdown or is shutting down. The Windows application must exit for CSF to shutdown.
- `CSF_SHUTDOWN2_RC`. This return code indicates that CSF has shutdown or is shutting down. The Windows application may continue and may attempt to reconnect to CSF. `ADX_CCSF_INIT` may be called to check the status of CSF.
- `CSF_APP_SHUTDOWN_RC`. This return code indicates that CSF encountered an unrecoverable error. The Windows application must exit so that CSF can shutdown.

By installing the DDS files required for programming, you will get the files required for your application to use the 4690 C APIs. These files are:

- **FDSCSFAP.H.** This file provides the prototype definitions for the 4690 C APIs. This file is in the FDS\INCLUDE\FDS subdirectory.
- **FDSCSFIN.H.** This file provides the definitions for the return codes coming from the ADX_CCSF_INIT and ADX_CCSF_DONTKILL APIs. This file is in the FDS\INCLUDE\FDS subdirectory.
- **FDSCSFOP.H.** This file provides the flag and option definitions for the 4690 C APIs. This file is in the FDS\INCLUDE\FDS subdirectory. **FDSPORT.H.** This file provides the type definitions needed for the 4690 C APIs. This file is in the FDS\INCLUDE\FDS subdirectory.
- **FDS4CAPI.LIB.** This file provides the necessary linkage information for your Windows applications to link with the 4690 C API support in CSF. This file is in the FDS\LIB subdirectory.

Wireless Communications Support

Symbol and Telxon controller to terminal wireless communications are supported by CSF.

New or Changed DDS/CSF Functions for Version 3

Windows XP and Windows Server 2003 Support

CSF Version 3 adds support for Windows XP Professional and Windows Server 2003. Windows NT Workstation, NT Server 4.0, Windows 2000 Pro, and Windows 2000 Workstation continue to be supported.

Automatic Installation of Java JVM

Upon installation of DDS/CSF Version 3, a fully compatible Java JVM is installed automatically. The customer has to do no Java configuration at all.

4690 OS Level Support

Support for 4690 OS V3R3 is added by CSF Version 3. 4690 OS V1 is not supported by CSF Version 3. Version 3 continues to support 4690 OS V2R3 and V2R4. See the DDS/CSF readme file (readme.fds) for specific details of what levels of 4690 OS are supported by CSF Version 3.

Additional Terminal Support

DDS/CSF Version 3 supports all terminals supported by the version of 4690 that you install. See above for the versions of 4690 that CSF now supports.

Store Loop Adapter Support

DDS/CSF Version 3 continues to support the use of both ISA and PCI store loop adapters in the controller for all versions of Windows.

CSF Support for SNA and ADCS Discontinued

DDS/CSF Version 3 does not support SNA communications or ADCS functions. Therefore, Microsoft's Host Integration System is no longer needed.

ASM under JAVA Discontinued

DDS/CSF does not support this function. The user should maintain his 4690 Images using a native 4690 system and distribute updates using Windows functions.

Artic Card Support Discontinued

DDS/CSF Version 3 no longer supports the use of the Artic card.

X.25 Communications Protocol Support Discontinued

DDS/CSF Version 3 no longer supports the use of this protocol which was associated with the Artic Card.

Maintaining Your 4690 OS Software

In a DDS enterprise that is using 4690 CSF, a 4690 OS library machine must be located at the central site. This 4690 controller has the 4690 OS installed on it, as well as any 4690 applications and features you intend to use.

Note: Refer to the README.FDS file located in the Disk1 folder of the DDS installation package to determine the level of 4690 OS that is supported.

The purpose of the 4690 OS library machine is to provide a source of 4690 OS software that can be distributed to your stores.

Once your stores have their Windows controllers with DDS, 4690 CSF, and 4690 OS, you must decide how to maintain the 4690 OS software in the store. You will receive DDS and 4690 CSF maintenance (in the form of CSDs or e-fixes) as Windows InstallShield packages. 4690 OS and 4690 OS application maintenance will be in the form of 4690 OS Apply Software Maintenance (ASM) diskettes. See the *4690 Store System: Planning, Installation, and Configuration Guide* for more information about the 4690 OS ASM diskettes.

DDS and 4690 CSF maintenance can be installed remotely using the DDS unattended installation process. 4690 OS and 4690 OS application maintenance, in the form of 4690 OS ASM diskettes, must be installed in a different way:

You should apply the 4690 OS software maintenance to your 4690 OS library machine at the central site and use any necessary Windows tools to update files at your remote DDS/CSF machines.

Note that DDS must be running on any DDS/CSF machine to allow writing of updated files to the controlled drive. If CSF is also running and has files open, there could be problems doing the update. It is recommended that CSF be stopped for the update. Note that, after the update is done on the primary controller, the updated files can be distributed via DDS to the backup controller.

Problem Determination for CSF and MCF

Both CSF and MCF capture data that is useful for debugging problems. This data is located in the LOG subdirectory of the DDS installation directory. These files are provided for use by QVS support and may be required for problem determination and resolution. The DDS utility FDSPADU can be used to collect the log files as well as DDS debugging information.

It is important to remember that the logged data is buffered in memory until CSF and MCF ends using the **FDSEPROC -k** command. This command can be entered manually, but the **net stop** command, a 4690OS Load Controller Storage, and a 4690OS Dump Controller Storage also cause the data to be written to the log files..

The FDSCLOG.BAT file preserves the last five sets of CSF log files with names CSFLOGS.00n, where n is a number from 1 to 5. Number 1 is the most recent. Additional information can be found in *QVS Distributed Data Services/Controller Services Feature for Windows Installation and Configuration Guide*.

CSF Initialization Messages

Refer to the *QVS Distributed Data Services/Controller Services Feature for Windows Installation and Configuration Guide* for CSF initialization messages.

Helpful Tips and Limitations

1. If the workstation on which you are running CSF contains a Cirrus Logic video card, you may experience situations where the display background changes to black with the icons showing as little boxes on the black field. Typically this happens when you press the Ctrl-Esc or the Alt-SysRq key combination. To clear this condition, press the key combination that caused the problem; you might have to do this more than once.
2. Although you can use Windows Task Manager to end CSF or DDS, errors or application dumps may occur. This is because normal application termination processes may not occur. You should only use Windows Task Manager to end CSF or DDS when the NET STOP command in Windows does not work.
3. RAM disks and CSF:
Four subdirectories are unique to CSF. They are located in the FDS subdirectory and are used to support the use of RAM disks by 4690 applications. The directories and their related 4690 RAM Disks are as follows:
 - 4690RAM1 T

- 4690RAM2 U
 - 4690RAM3 V
 - 4690RAM4 W
4. Node role changes and the effect on CSF:
 DDS has a primary role and a backup role. You can use the Node Control Utility to change these. When the DDS primary node role is deactivated, this action affects CSF.

When the master or file server role is deactivated in a native 4690 system, 4690 stops any background applications that were started automatically, because the controller was the master or file server. 4690 also stops any background applications that were configured to stop when the controller is not the master or file server. These actions take place before the node role is physically deactivated at the lowest level.

With CSF, the actions are the same, but the order of actions is different. The request to deactivate the primary role is made directly to DDS. It is not made through CSF. Consequently, the node role deactivation takes place before CSF is notified of the deactivation. When the primary node role is deactivated, distributed 4690 files that the applications have open for write access become unavailable. These applications may log 4690 events about these errors during the window of time when the primary node role is deactivated, but CSF has not stopped the background applications.

To summarize, background 4690 applications may log 4690 events regarding files being inaccessible or unavailable after DDS has been requested to deactivate the primary-node role. These 4690 events should be expected in this situation.

5. Printer setup: 4690 printer devices are mapped to Windows as follows:
- PRN1:, LPT1: maps to LPT1
 - PRN2:, LPT2: maps to LPT2
 - PRN3:, LPT3: maps to LPT3

4690 device drivers can be defined as serial or parallel.

There are two methods in Windows to enable device LPT1 to print, other than using a physical, parallel port:

- a. Printer Pooling method: Windows supports a concept called pooled printers. With this method, Windows spreads print jobs across the pool of printers. You can define LPT1 and a network printer to be in the same pool. If no printer is attached to LPT1, Windows sends the print job to the network printer because LPT1 is in an error state and cannot be used.
- b. Net Use method: If you want LPT1 to go to a network printer, use the following command. “NET USE LPT1: \\srv1\printer”, where srv1 is

the name of the server to which the printer is attached and printer is the name of the printer.

Appendix A. Using the Keyboard Keys with DDS Utilities

You can use either a mouse or the keyboard to perform tasks in the DDS utilities. This section describes how to use the keyboard.

Selecting a Menu Bar Option

The menu bar for each DDS Utility provides access to the functions associated with that utility.

1. Use one of the following procedures to select the applicable menu bar item and display its pull-down menu:
 - Press **F10** to select the menu bar. Use the left and right arrow keys to move across the menu bar and select the item you want. Then, press **Enter** or the down arrow key.
 - Press **F10** to select the menu bar. Press the underscored letter in the name of the menu bar item. For example, you can display the Administration pull-down menu from within the Node Control Utility by typing **A**.
 - Press **Alt** and the underscored letter in the name of the desired menu-bar item.
2. Use one of the following procedures to select an option on the pull-down menu:
 - Use the up and down arrow keys to select the option you want, and then press **Enter**.
 - Press the underscored letter in the name of the pull-down menu option.
 - If a menu item has a cascaded menu with additional choices, a right-pointing arrow appears to the right of the menu item. Either of the other methods for selecting an option displays the cascaded menu. You can also use the up and down arrow keys to select the option you want, and then press the right arrow key.
3. If there is a cascaded menu, use either of the following procedures to select the operation that you want to perform on the cascaded menu:
 - Use the arrow keys to select the operation that you want to perform on the cascaded menu and then press **Enter**.
 - Press the underscored letter in the name of the cascaded menu item. For example, press **N** to select New Configuration File under the Generate Configuration File menu option of the Configuration Utility.

If a menu item requires that you enter some information to continue, the ellipsis symbol (...) is included in the menu item.

Function Keys and Keyboard Keys

A list of the function keys available in the current operation appears at the bottom of every panel. You perform the function by pressing the corresponding key.

Keyboard keys, unlike function keys, do not appear on the screen. They are keys on the system keyboard that are labeled with actions. Examples of these keys are Enter and Tab.

Function Keys

Table 1. Function Keys

Key	Name	Description
F1	Help	Accesses a standard help facility that can be opened and used at any time. Help provides guidance and useful information to help use the application. To exit from Help, press the Esc key.
F11	Accept	Tells the utility to accept the entered data.
Esc	Escape	Cancel the current process and returns to the previous step.

Keyboard Keys

Table 2. Function Keys

Key	Name	Description
Backspace	Backspace	Deletes one character to the left of the cursor.
Shift+Tab	Back tab	Moves the cursor to the previous field, except in Data, where it moves the cursor to the top of the screen.
Ctrl+Home	Beginning of data	Moves the cursor to the top leftmost position in the current field.
Home	Beginning of line	Moves the cursor to the leftmost choice in a group of choices, or to the beginning of the current line in an entry field.
Esc	Escape	Cancel the current process and returns to the previous step.
Enter	Default action/ New line	Accepts the default for the control that is selected and moves to the next field or control.
Ctrl+Enter		Saves the data in the current field and moves the cursor to the previous field.
Delete	Delete	Removes the selected character or group of characters from view and compresses the space previously occupied by the character or group of characters.
Ctrl+End	End of data	Moves the cursor to the bottom, rightmost position in the current field.
End	End of line	Moves the cursor to the rightmost choice in a group of choices, or to the end of the current line in an entry field.
Arrow keys	Move cursor	Moves the cursor left, right, up, or down. At the last choice in a direction within a group of buttons, the cursor wraps.

Table 2. Function Keys (continued)

Key	Name	Description
Ctrl+Page Up	Page left	Scrolls to display information to the left of the currently visible window area.
Ctrl+Page Down	Page right	Scrolls to display information to the right of the currently visible window area.
space bar	Select / Space	In a list where more than one item can be selected, selects the item or choice on which the cursor is positioned. In an entry field, creates a space character.
Insert	Insert / Replace	Toggles between insert and replace modes in text entry.
Alt+Esc	Switch between windows not associated	Switches to the next open window, full-screen session, or icon that is minimized on the desktop.
Shift+Esc	Display the system menu	Displays the system menu in the upper left corner of the screen.
Tab	Tab	Moves the cursor to the next field. The cursor is positioned either on the first choice in that field or on the currently set choice in that field. The cursor moves from left to right and top to bottom. At the bottom rightmost field, the cursor wraps to the top leftmost field.
Ctrl+Esc	Window list	Displays the window list.
Ctrl+Left Arrow	Word left	Moves the cursor to the beginning of the word to the left of the cursor.
Ctrl+Right Arrow	Word right	Moves the cursor to the end of the word to the right of the cursor, including any trailing space characters.

Appendix B. Hard Disk Failure Recovery

Use the following procedures to recover from hard disk failures. The recovery procedure is dependent on the role the workstation assumes in DDS.

1. If the hard drive on the acting primary distributor failed, activate the acting backup distributor as the acting primary distributor.
2. Replace or repair the hard disk.
3. Install Windows, if the old hard disk contained one of these operating systems.
4. Install and configure DDS, if the old hard disk contained DDS.
5. Issue the command: **FDSBKDR dim delete**.

The following steps are required to recover distributed data and are specific to the role of the workstation:

Primary Distributor

Start DDS and allow the workstation to reconcile. When completely reconciled, deactivate the acting primary distributor and activate the workstation as the acting primary distributor. If your system had no acting backup distributor role, the workstation cannot be reconciled. Information must be loaded into a controlled drive and distribution must be set for the files or subdirectories to be distributed.

Backup Distributor

Start DDS and allow the workstation to reconcile. When reconciling is complete, the workstation automatically assumes the acting backup distributor role. No other action is required.

Subordinate Workstation

Start DDS and allow the workstation to reconcile. No other action is required.

Appendix C. Gathering Problem Determination Data

This section provides suggestions on collecting problem determination data when you encounter a problem. You may want to consider creating a command file to automate the steps described in the following section. You can place a shortcut to the command file on the desktop to enable end users to more easily gather problem data. You should gather problem documentation as soon as you suspect a problem. Collect standard problem determination data, write a detailed description of what operations led up to the problem, and create a description of the problem symptoms as soon as possible after you encounter a problem so that no data is lost and the problem details can be recorded accurately.

To gather problem determination data, perform the following steps:

1. From an NT command prompt on the acting primary controller, issue `FDSADMIN /DUMP *`.

This will generate a DDS dump file on all active DDS workstations communicating with the acting primary controller.

2. For each controller click **Start** from the taskbar, point to Programs, point to **QVS DDS**, and click **Problem Analysis Utility** to run the Problem Analysis Utility. You can also type `FDSPADU` and press **Enter** from the command line to start the utility.

Instead of using the Problem Analysis Utility, you can also type `FDSPADC` on the command line to run the command line version. Make sure that you run this on the Acting Primary Controller, the Acting Backup Controller, and a Subordinate (if the problem encountered involves a DDS subordinate).

Note: It is very important to always collect problem data from both controllers, due to the high amount of interaction between the controllers.

3. Submit problem data from step 2, which generated a file named `FDSSAVE.FIL` in a directory named `FDSSAVEx` (where x is from 0 to 9) on the drive you specified in step 2.

For more information on the Problem Analysis Utility, see [Chapter 4. Problem Analysis Utility](#) on page 31.

Notices

This information was developed for products and services offered in the U.S.A. QVS may not offer the products, services, or features discussed in this document in other countries. Consult your local QVS representative for information on the products and services currently available in your area.

References in this publication to QVS products, programs, or services do not imply that QVS intends to make these available in all countries in which QVS operates. Any reference to a QVS product, program, or service is not intended to state or imply that only QVS's product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of QVS's intellectual property rights may be used instead of the QVS product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by QVS, are the user's responsibility.

QVS may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

QVS Director of Licensing
QVS Software, Inc.
5711 Six Forks Rd, Suite 300
Raleigh, NC 27609 U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the QVS Intellectual Property Department in your country or send inquiries, in writing, to:
QVS Director of Licensing
QVS Software, Inc.
5711 Six Forks Rd, Suite 300
Raleigh, NC 27609 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: QVS SOFTWARE, INC. PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. QVS may make improvements and/or changes in the product(s) and/or program(s) described in this publication at any time without notice.

Any references in this information to non-QVS Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this QVS product and use of those Web sites is at your own risk.

Licenses of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

QVS Software, Inc.
5711 Six Forks Rd, Suite 300
Raleigh, NC 27609 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee. The licensed program described in this document and all licensed material available for it are provided by QVS under terms of the QVS Customer Agreement, QVS International Program License Agreement, or any equivalent agreement between us.

Information concerning non-QVS products was obtained from the suppliers of those products, their published announcements or other publicly available sources. QVS has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-QVS products. Questions on the capabilities of non-QVS products should be addressed to the suppliers of those products.

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

IBM
StorePlace
System/370
VisualAge

NetView is a trademark of Tivoli Systems Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, Windows 2000, Windows XP, and Windows Server 2003 are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java is a trademark of Sun Microsystems, Inc. in the United States, other countries, or both. Other company, product, and service names may be trademarks or service marks of others.

Glossary

This glossary defines terms and abbreviations used in this manual. It includes terms and definitions from the *IBM Dictionary of Computing* (New York; McGraw-Hill, Inc., 1994).

The following cross-references are used in this glossary:

- **Contrast with.** This refers to a term that has an opposed or substantively different meaning.
- **See.** This refers the reader to multiple-word terms in which this term appears.
- **See also.** This refers the reader to terms that have a related, but not synonymous, meaning.
- **Synonym for.** This indicates that the term has the same meaning as a preferred term, which is defined in the glossary.

A

active. (1) Able to communicate on the network. A token-ring network adapter is active if it is able to transmit and receive on the network. (2) Operational. (3) Pertaining to a node or device that is connected or is available for connection to another node or device. (4) Currently transmitting or receiving.

adapter. (1) In the point-of-sale terminal, a circuit card that, with its associated software, enables the terminal to use a function or feature. (2) In a LAN, within a communicating device, a circuit card that, with its associated software and/or microcode, enables the device to communicate over the network.

address. (1) In data communication, the IEEE-assigned unique code or the unique locally administered code assigned to each device or workstation connected to a network. (2) A character, group of characters, or a value that identifies a register, a particular part of storage, a data source, or a data link. The value is represented by one or more characters. (3) To refer to a device or an item of data by its address. (4) The location in the storage of a computer where data is stored.

advanced program-to-program communications (APPC). An implementation of the SNA/SDLC LU 6.2 protocol that allows interconnected systems to communicate and share the processing of programs.

alert. (1) An error message sent to the system services control point (SSCP) at the host system. (2) For IBM LAN management products, a notification indicating a possible security violation, a persistent error condition, or an interruption or potential interruption in the flow of data around the network. See also network management vector transport. (3) In SNA, a record sent to a system problem management focal point to communicate the existence of an alert condition. (4) In the NetView program, a high-priority event that warrants immediate attention. This data base record is generated for certain event types that are designed by user-constructed filters.

alias. An alternate name by which a named entity can be known on a network. See also **name**.

American National Standard Code for Information Interchange (ASCII). The standard code, using a coded character set consisting of 7-bit coded characters (8 bits including parity check), used for information interchange among data processing systems, data communication systems, and associated equipment. The ASCII set consists of control characters and graphics characters.

API. Application program interface.

APPC. Advanced program-to-program communications.

application. (1) A collection of one or more programs that work together to accomplish goals for a business. (2) A set of executable files and data files required to perform a desired function, which can consist of multiple programs running on different workstations.

application program. (1) A program written for or by a user that applies to the user's own work. (2) A program written for or by a user that applies to a particular application. (3) A program used to connect and communicate with stations in a network, enabling users to perform application-oriented activities.

application program interface (API). The formally defined programming language interface that is between an QVS system control program or a licensed program and the user of the program.

ASCII. American National Standard Code for Information Interchange.

attach. (1) To connect a device physically. (2) To make a device a part of a network logically. Compare with connect.

attaching device. Any device that is physically connected to a network and can communicate over the network.

B

background. On a color display, the part of the display screen that surrounds a character.
backup. Pertaining to a system, device, file, or facility that can be used in the event of a malfunction or the loss of data.

batch. Smaller subdivisions of price change records within an event. Each batch has a 12-character ID and a 30-character description field.

binary. (1) Pertaining to a system of numbers to the base two; the binary digits are 0 and 1. (2) Pertaining to a selection, choice, or condition that has two possible different values or states.

bit. Either of the binary digits: 0 or 1.

block size. (1) The minimum size that frames are grouped into for retransmission. (2) The number of data elements (such as bits, bytes, characters, or records) that are recorded or transmitted as a unit.

bridge. (1) An attaching device connected to two LAN segments to allow the transfer of information from one LAN segment to the other. A bridge may connect the LAN segments directly by network adapters and software in a single device, or may connect network adapters in two separate devices through software and use of a telecommunications link between the two adapters. (2) A functional unit that connects two LANs that use the same logical link control (LLC) procedures but may use the same or different medium access control (MAC) procedures. Contrast with gateway and router.

broadcast. Simultaneous transmission of data to more than one destination.

broadcast domain. In the QVS Distributed Data Services, a group of nodes within a system. It can be as small as zero nodes (an empty broadcast domain). It can also include every node in the system. Broadcast domains have unique broadcast domain names. The Data Distribution component supports distributing files to all nodes within a broadcast domain. The Interprocess Communications component can send a message to every node within a broadcast domain.

byte. (1) A string that consists of a number of bits, treated as a unit, and representing a character. (2) A binary character operated upon as a unit and usually shorter than a computer word. (3) A string that consists of a particular number of bits, usually 8, that is treated as a unit, and that represents a character. (4) A group of 8 adjacent binary digits that represent one extended binary-coded decimal interchange code (EBCDIC). See n-bit byte.

C

C. A high-level programming language designed to optimize run time, size, and efficiency.

code page. A particular assignment of hexadecimal identifiers to graphic characters.

call. The action of bringing a function or subprogram into effect, usually by specifying the entry conditions and jumping to an entry point.

chain. (1) Transfer of control from the currently executing program to another program or overlay. (2) Referencing a data record from a previous data record.

chaining. A method of storing records in which each record belongs to a list or group of records and has a linking field for tracing the chain.

chaining threshold. The number of chains in a keyed file that causes a message to be logged by the operating system.

checkpoint. A point at which information about the status of a job and the system can be recorded so that the job step can be restarted later.

clear. To delete data from a screen or from memory.

command. (1) A request for performance of an operation or execution of a program. (2) A character string from a source external to a system that represents a request for system action.

Common User Access (CUA). The portion of the IBM Systems Application Architecture (SAA) that defines the basic elements of the end-user interface and how to use them. These elements include screen layout, menu presentation and selection techniques, keyboard layout and use, and display options.

component. (1) Any part of a network other than an attaching device, such as an IBM 8228 Multistation Access Unit. (2) Hardware or software that is part of a functional unit.

concatenate. To join one string to another.

configuration. (1) The group of devices, options, and programs that make up a data processing system or network as defined by the nature, number, and chief characteristics of its functional units. More specifically, the term may refer to a hardware configuration or a software configuration. See also system configuration.

(2) In the QVS Distributed Data Services, program options that are initially set at installation, and that can be changed later. Changing these options requires an IPL. These changes must be performed by a programmer or store operations personnel. These options are used to tune the product's use of the operating system and machine resources.

configuration file. The collective set of definitions that describes a configuration.

connect. In a LAN, to physically join a cable from a station to an access unit or network connection point. Contrast with attach.

constant. String or numeric value that does not change throughout program execution.

controlled drive. In the QVS Distributed Data Services, hard disk partitions that are controlled by the QVS Distributed Data Services. Controlled drives can benefit from disk I/O prioritization and can store either the prime copy or the image copy of distributed data. Distributed files must reside on a controlled drive.

controller. A unit that controls input/output operations for one or more devices.

conversation state. The condition of a conversation that reflects what the past action on that conversation has been and that determines what the next set of actions may be.

D

data. (1) A representation of facts, concepts, or instructions in a formalized manner suitable for communication, interpretation, or processing by human or automatic means. (2) Any representations such as characters or analog quantities to which meaning is or might be assigned.

data file. A collection of related data records organized in a specific manner; for example, a payroll file (one record for each employee, showing such information as rate of pay and deductions) or an inventory file (one record for each inventory item, showing such information as cost, selling price, and number in stock.) See also data set, file.

datagram. (1) A particular type of information encapsulation at the network layer of the adapter protocol. No explicit acknowledgment for the information is sent by the receiver. Instead, transmission relies on the “best effort” of the link layer. (2) A self-contained packet, independent of other packets, that carries information sufficient for routing from the originating node to the destination node without relying on earlier exchanges between the nodes and the network.

data link. (1) Any physical link, such as a wire or a telephone circuit, that connects one or more remote terminals to a communication control unit, or connects one communication control unit with another. (2) The assembly of parts of two data terminal equipment (DTE) devices that are controlled by a link protocol, and the interconnecting data circuit, that enable data to be transferred from a data source to a data link. (3) In SNA, see also link.

Note: A telecommunication line is only the physical medium of transmission. A data link includes the physical medium of transmission, the protocol, and associated devices and programs; it is both physical and logical.

data processing system. A network, including computer systems and associated personnel, that accepts information, processes it according to a plan, and produces the desired results.

data set. Logically related records treated as a single unit. See also file.

deactivation. The process of taking any element out of service, rendering it inoperative, or placing it in a state in which it cannot perform the functions for which it was designed.

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

default value. The value the system supplies when the user does not specify a value.

destination. Any point or location, such as a node, station, or particular terminal, to which information is to be sent.

device. (1) A mechanical, electrical, or electronic contrivance with a specific purpose. (2) An input/output unit such as a terminal, display, or printer. See also attaching device.

device driver. The code needed to attach and use a device on a computer or a network.

directory. (1) A table of identifiers and references that correspond to items of data. (2) An index that a control program uses to locate one or more blocks of data that are stored in separate areas of a data set in direct access storage.

directory services. Services for resolving user identifications of network components to network routing information.

disabled. (1) Pertaining to a state of a processing unit that prevents the occurrence of certain types of interruptions. (2) Pertaining to the state in which a transmission control unit or audio response unit cannot accept incoming calls on a line.

disk. A round, flat plate coated with a magnetic substance on which computer data is stored. See also integrated disk, hard disk.

diskette. A thin, flexible magnetic disk permanently enclosed in a protective jacket. A diskette is used to store information for processing.

diskette drive. The mechanism used to seek, read, and write data on diskettes.

disk operating system (DOS). An operating system for computer systems that use disks and diskettes for auxiliary storage of programs and data.

display. (1) A visual presentation of data. (2) A device that presents visual information to the point-of-sale terminal operator and to the customer, or to the display station operator.

distributed. Physically separate but connected by cables.

distribution directory. In the QVS Distributed Data Services, a directory of information that determines which files are distributed, which nodes the files are distributed to, and the distribution frequency for each file. Each entry in the distribution directory represents either a single file or a subdirectory.

distribution domain. In the QVS Distributed Data Services, a group of nodes that have data distribution capability. There are two types of distribution domains: mirrored domains and broadcast domains.

distribution frequency. In the QVS Distributed Data Services, the distribution frequency of a file is specified in the distribution directory. There are two types of distribution frequency: distribute on update and distribute on close.

distribution role. In the QVS Distributed Data Services, defines the capabilities of a node in relation to whether the prime copy or an image copy of a file is controlled. There are three possible distribution roles: primary distributor, backup distributor, and subordinate.

domain. An SSCP and the resources that it can control.

DOS. Disk operating system.

driver. Software component that controls a device.

dump. (1) To write at a particular instant the contents of storage, or part of storage, onto another data medium for the purpose of safeguarding or debugging the data. (2) Data that has been dumped.

E

enabled. (1) On a LAN, pertaining to an adapter or device that is active, operational, and able to receive frames from the network. (2) Pertaining to a state of a processing unit that allows the occurrence of certain types of interruptions. (3) Pertaining to the state in which a transmission control unit or an audio response unit can accept incoming calls on a line.

error condition. The condition that results from an attempt to use instructions or data that is invalid.

error message. A message that is issued because an error has been detected.

evaluation. Reduction of an expression to a single value.

event. (1) Processing unit containing price changes and item file updates. All records in an event share common characteristics such as type of change and event due date. (2) An occurrence of significance to a task; for example, the completion of an asynchronous operation, such as an I/O operation.

exception. An abnormal condition such as an I/O error encountered in processing a data set or a file. See also overflow exception and underflow exception.

execution. The process of carrying out an instruction or instructions of a computer program by a computer.

exit. To execute an instruction or statement within a portion of a program in order to terminate the execution of that portion.

Note: Such portions of programs include loops, routines, subroutines, and modules.

F

feature. A part of a QVS product that may be ordered separately by the customer.

FCS (Frame check sequence). (1) A character determined by the data present within the transmission frame and appended to the transmission frame to allow detection of transmission errors. (2) A field immediately preceding the closing flag sequence of a frame

that contains a bit sequence checked by the receiver to detect transmission errors. (3) In SDLC, 16 bits in a frame that contain transmission-checking information.

field. On a data medium or a storage medium, a specified area used for a particular category of data; for example, a group of character positions used to enter or display wage rates on a panel.

file. A named set of records stored or processed as a unit. For example, an invoice may form a record and the complete set of such records may form a file. See also data file and data set.

file access. Methods of entering a file to retrieve the information stored in the file.

file extension. Extension to a file name. A file extension is optional and can be up to three characters long. All characters permitted in a file name are permitted in a file extension. The file extension must be separated from the file name by a period.

file name. (1) A name assigned or declared for a file. (2) The name used by a program to identify a file.

file specification. (1) Unique file identifier. A file specification includes an optional drive specification followed by a colon, a primary file name of one to eight characters, and an optional period and file type of zero to three characters. (2) In the QVS Distributed Data Services, an operating system fully-qualified path name for a file on a node that uniquely identifies it on that particular node.

file type. The attribute of a file that specifies to which store controllers it is distributed.

fixed disk (drive). Synonym for hard disk.

frame. (1) The unit of transmission in some LANs, including the IBM Token-Ring Network and the IBM PC Network. It includes delimiters, control characters, information, and checking characters. On a token-ring network, a frame is created from a token when the token has data appended to it. On a token-bus network (IBM PC Network), all frames including the token frame contain a preamble, start delimiter, control address, optional data and checking characters, end delimiter, and are followed by a minimum silence period. (2) A housing for machine elements. (3) In synchronous data link control (SDLC), the vehicle for every command, every response, and all information that is transmitted using SDLC procedures. Each frame begins and ends with a flag.

frequency. The rate of signal oscillation, expressed in hertz (cycles per second).

function. (1) A specific purpose of an entity, or its characteristic action. (2) A subroutine that returns the value of a single variable. (3) In data communications, a machine action such as a carriage return or line feed.

G

gateway. A device and its associated software that interconnect networks of systems of different architectures. The connection is usually made above the Reference Model network layer. For example, a gateway allows LANs access to System/370® host computers. Contrast with bridge and router.

group. (1) A set of related records that have the same value for a particular field in all records. (2) A collection of users who can share access authorities for protected resources. (3) A list of names that are known together by a single name.

H

hard disk (drive). In a personal computer system unit, a disk storage device that reads and writes on rigid magnetic disks. It is faster and has a larger storage capacity than a diskette and is permanently installed.

hardware. Physical equipment as opposed to programs, procedures, rules, and associated documentation.

hashing. In an indexed data set, using an algorithm to convert the key of a record to an address for that record, for storing and retrieving data. Synonym for randomizing.

header. The portion of a message that contains control information for the message such as one or more destination fields, name of the originating station, input sequence number, character string indicating the type of message, and priority level for the message.

home record. The first record in a chain of records.

I

IBM Disk Operating System (DOS). A disk operating system based on MS-DOS.
identifier. String of characters used to name elements of a program, such as variable names, reserved words, and user-defined function names.

IEEE. Institute of Electrical and Electronics Engineers.

inactive. (1) Not operational. (2) Pertaining to a node or device not connected or not available for connection to another node or device. (3) In the IBM Token-Ring Network, pertaining to a station that is only repeating frames or tokens, or both.

information (I) frame. A frame in I format used for numbered information transfer. See also supervisory frame, unnumbered frame.

initialize. In a LAN, to prepare the adapter (and adapter support code, if used) for use by an application program.

interprocess communication. (1) The exchange of information between processes or threads through semaphores, queues, and shared memory. (2) A component of QVS

Distributed Data Services that provides a peer-to-peer messaging services between processes running on a single node (intranode IPC) and on different nodes (internode IPC).

IPC. Interprocess communication.

initial program load (IPL). The initialization procedure that causes an operating system to begin operation.

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.

insert. To make an attaching device an active part of a LAN.

instance. In the QVS Distributed Data Services, a copy of a distributed file on a given node within a distribution domain. There are three types of instances: prime copy, backup copy, and image copy.

integrated disk. An integral part of the processor that is used for magnetically storing files, application programs, and diagnostics. Synonym for disk.

interactive. Pertaining to an application or program in which each entry calls forth a response from a system or program. An interactive program may also be conversational, implying a continuous dialog between the user and the system.

interface. (1) A shared boundary between two functional units, defined by functional characteristics, common physical interconnection characteristics, signal characteristics, and other characteristics as appropriate. (2) A shared boundary. An interface may be a hardware component to link two devices or a portion of storage or registers accessed by two or more computer programs. (3) Hardware, software, or both, that links systems, programs, or devices.

I/O. Input/output.

IPL. Initial program load. See also reboot.

item. (1) One member of a group. (2) In a store, one unit of a commodity, such as one box, one bag, or one can. Usually an item is the smallest unit of a commodity to be sold.

K

K. When referring to storage capacity, a symbol that represents two to the tenth power, or 1024.

keyboard. A group of numeric keys, alphabetic keys, special character keys, or function keys used for entering information into the terminal and into the system.

keyed file. (1) Type of file composed of keyed records. Each keyed record has two parts: a key and data. A key is used to identify and access each record in the file. (2) In the QVS Distributed Data Services, permanent files that can be either local or remote to the application. Access to keyed files is based on a key field that is situated at the front of each keyed file record. All records within a keyed file must have the same length. A common example in the retail industry is the item price lookup file.

L

LAN. Local area network.

LAN adapter. The circuit card within a communicating device (such as a personal computer) that, together with its associated software, enables the device to be attached to a LAN.

line. On a terminal, one or more characters entered before a return to the first printing or display position.

link. (1) In the IBM Store System, the logical connection between nodes including the end-to-end link control procedures. (2) The combination of physical media, protocols, and programming that connects devices on a network. (3) In computer programming, the part of a program, in some cases a single instruction or an address, that passes control and parameters between separate portions of the computer program. (4) To interconnect items of data or portions of one or more computer programs. (5) In SNA, the combination of the link connection and link stations joining network nodes. See also link connection.

Note: A link connection is the physical medium of transmission; for example, a telephone wire or a microwave beam. A link includes the physical medium of transmission, the protocol, and associated devices and programming; it is both logical and physical.

link connection. (1) All physical components and protocol machines that lie between the communicating link stations of a link. The link connection may include a switched or leased physical data circuit, a LAN, or an X.25 virtual circuit. (2) In SNA, the physical equipment providing two-way communication and error correction and detection between one link station and one or more other link stations. (3) In the IBM Store System, the logical link providing two-way communication of data from one network node to one or more other network nodes.

load. In computer programming, to enter data into memory or working registers.

local area network (LAN). A computer network located on a user's premises within a limited geographical area.

Note: Communication within a LAN is not subject to external regulations; however, communication across the LAN boundary may be subject to some form of regulation.

local file. In the QVS Distributed Data Services, a file that is not distributed (is not a prime copy, backup copy, or image copy). All files on a node are local files, by default. Contrast with remote file.

local node. The node on which an QVS Distributed Data Services function or API that is being described is running. Contrast with remote node.

logical name. In the QVS Distributed Data Services, user-defined names that can be substituted for hard-coded file names, interprocess communication queue names, and node IDs. Logical names are dynamically resolved when an application runs. Logical names can also be used to identify roles. See also alias.

logon (n). The procedure for starting up a point-of-sale terminal or store controller for normal sales operations by sequentially entering the correct security number and transaction number. Synonym for sign-on.

M

media. Plural form of medium.

medium. (1) A physical carrier of electrical or optical energy. (2) A physical material in or on which data may be represented.

megabyte (MB). (1) For processor storage and real and virtual memory, two to the twentieth power or 1,048,576 bytes in decimal notation. (2) For disk storage capacity and transmission rates, ten to the sixth power or 1,000,000 in decimal notation.

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

message. (1) An arbitrary amount of information whose beginning and end are defined or implied. (2) A group of characters and control bit sequences transferred as an entity. (3) In telecommunication, a combination of characters and symbols transmitted from one point to another. (4) A logical partition of the user device's data stream to and from the adapter. See also error message, operator message.

N

name. An alphanumeric term that identifies a data set, statement, program, or cataloged procedure.

n-bit byte. A string that consists of n bits.

network. (1) A configuration of data processing devices and software connected for information interchange. (2) An arrangement of nodes and connecting branches. Connections are made between data stations.

network management vector transport (NMVT). The portion of an alert transport frame that contains the alert message.

node. (1) Any device, attached to a network, that transmits and/or receives data. (2) An end point of a link, or a junction common to two or more links in a network. Nodes can be processors, controllers, or workstations. Nodes can vary in routing and other functional capabilities. (3) In a network, a point where one or more functional units interconnect transmission lines. (4) In the QVS Distributed Data Services, a LAN-attached machine that is running the QVS Distributed Data Services. Synonym for workstation.

node ID. The identifier for an QVS Distributed Data Services node. Each node has a single node ID, regardless of how many LANs are used to connect it to other nodes. Node IDs are specified during the installation of the QVS Distributed Data Services. A node ID must be unique within a single QVS Distributed Data Services system. Node IDs cannot be changed without reinstalling the QVS Distributed Data Services.

O

office. See work area.

offline. Operation of a functional unit without the control of a computer or control unit.

online. Operation of a functional unit that is under the continual control of a computer or control unit. The term also describes a user's access to a computer using a terminal.

open. (1) To make an adapter ready for use. (2) A break in an electrical circuit. (3) To make a file ready for use.

operating system. Software that controls the execution of programs. An operating system may provide services such as resource allocation, scheduling, input/output control, and data management. Examples are IBM DOS and IBM OS/2®.

operation. (1) A defined action, namely, the act of obtaining a result from one or more operands in accordance with a rule that completely specifies the result for any permissible combination of operands. (2) A program step undertaken or executed by a computer. (3) An action performed on one or more data items, such as adding, multiplying, comparing, or moving.

operator. (1) A symbol that represents the action being performed in a mathematical operation. (2) A person who operates a machine.

operator message. A message from the operating system or a program telling the operator to perform a specific function or informing the operator of a specific condition within the system, such as an error condition.

option. (1) A specification in a statement, a selection from a menu, or a setting of a switch, that may be used to influence the execution of a program. (2) A hardware or software function that may be selected or enabled as part of a configuration process. (3) A piece of hardware (such as a network adapter) that can be installed in a device to modify or enhance device function.

OS. Operating system.

overflow exception. A condition caused by the result of an arithmetic operation having a magnitude that exceeds the largest possible number. See also underflow exception.

P

page. (1) The portion of a panel that is shown on a display surface at one time. (2) To move back and forth among the pages of a multiple-page panel. See also scroll. (3) In a virtual storage system, a fixed-length block that has a virtual address and is transferred as a unit between main storage and auxiliary storage.

panel. The complete set of formatted information that appears in a single display on a visual display unit.

parameter. (1) A name in a procedure that is used to refer to an argument passed to that procedure. (2) A variable that is given a constant value for a specified application and that may denote the application. (3) An item in a menu or for which the user specifies a value or for which the system provides a value when the menu is interpreted. (4) Data passed between programs or procedures.

path. (1) Reference that specifies the location of a particular file within the various directories and subdirectories of a hierarchical file system. (2) In a network, any route between any two nodes. (3) The route traversed by the information exchanged between two attaching devices in a network. (4) A command in Windows that specifies directories to be searched for commands or batch files that are not found by a search of the current directory.

personal computer (PC). A desk-top, free-standing, or portable microcomputer that usually consists of a system unit, a display, a keyboard, one or more diskette drives, internal fixed-disk storage, and an optional printer. PCs are designed primarily to give independent computing power to a single user and are inexpensively priced for purchase by individuals or small businesses. Examples include the various models of the IBM Personal Computers.

pointer. (1) An identifier that indicates the location of an item of data in memory. (2) A data element that indicates the location of another data element. (3) A physical or symbolic identifier of a unique target.

point-of-sale terminal. (1) The IBM 4683 Point of Sale Terminal: A unit that provides point-of-sale transaction, data collection, credit authorization, price look-up, and other inquiry and data entry functions. (2) The IBM 4683/4684 Point of Sale Terminal: A unit that provides central processing, point-of-sale transaction, data collection, credit authorization, price look-up, and other inquiry and data entry functions.

polling characters (address). A set of characters specific to a terminal and the polling operation; response to these characters indicates to the computer whether the terminal has a message to enter.

port. (1) An access point for data entry or exit. (2) A connector on a device to which cables for other devices such as display stations and printers are attached. Synonym for socket.

primary distributor. In the QVS Distributed Data Services, a node that can control the prime copies of all distributed files.

problem determination. The process of determining the source of a problem as being a program component, a machine failure, a change in the environment, a common-carrier link, a user-supplied device, or a user error.

programmable power. The function that provides the ability to turn power on and off at workstations remotely, without physical intervention.

prompt. A character or word displayed by the operating system to indicate that it is ready to accept input.

protocol. (1) A set of semantic and syntactic rules that determines the behavior of functional units in achieving communication. (2) In SNA, the meanings of and the sequencing rules for requests and responses used for managing the network, transferring data, and synchronizing the states of network components. (3) A specification for the format and relative timing of information exchanged between communicating parties.

Q

queue. A line or list formed by items in a system waiting for service; for example, tasks to be performed or messages to be transmitted in a message routing system.

queue name. An identifier for a queue that uniquely identifies it on a particular node.

R

randomizing. Synonym for hashing.

randomizing divisor. When creating a keyed file, a number used to calculate the number of sectors that can contain data in a keyed file. It is usually the largest prime number less

than the total number of sectors in the file. When a record is searched for, the randomizing divisor is used to calculate the location of the record.

read. To acquire or to interpret data from a storage device, from a data medium, or from another source.

reboot. To reinitialize the execution of a program by repeating the initial program load (IPL) operation.

receive. To obtain and store information transmitted from a device.

reconciliation. In the QVS Distributed Data Services, the process of making an image copy of a file identical to the prime copy. There are two forms of reconciliation: full reconciliation and partial reconciliation.

record. A collection of related items of data, treated as a unit; for example, in stock control, each invoice could constitute one record. A complete set of such records may form a file.

register. (1) A storage area in a computer's memory where specific data is stored. Registers are used in the actual manipulation of data values during the execution of a program. (2) A storage device having a specified storage capacity such as bit, byte, or computer word, and usually intended for a special purpose. (3) In the IBM Store System, a term that refers to the point-of-sale terminal.

REM. Ring error monitor.

remote file. In the QVS Distributed Data Services, a file that exists on a remote node. Contrast with local file.

remote node. In the QVS Distributed Data Services, any node within a system other than the local node. Contrast with local node.

remove. (1) To take an attaching device off a network. (2) To stop an adapter from participating in data passing on a network.

response. The information the network control program sends to the access method, usually in answer to a request received from the access method. (Some responses, however, result from conditions occurring within the network control program, such as accumulation of error statistics.)

retail path specification. An identifier for a file or queue consisting of the file or queue name, the operating system path, and (optionally) the node ID or broadcast domain name.

return code. (1) A value (usually hexadecimal) provided by an adapter or a program to indicate the result of an action, command, or operation. (2) A code used to influence the execution of succeeding instructions. (3) A value established by the programmer to be used

to influence subsequent program action. This value can be printed as output or loaded in a register.

ring error monitor (REM). A function that compiles error statistics reported by adapters on a network, analyzes the statistics to determine probable error cause, sends reports to network manager programs, and updates network status conditions. It assists in fault isolation and correction.

role. For the QVS Distributed Data Services, see distribution role.
role name. For the QVS Distributed Data Services, a logical name that is used specifically to identify a role.

root directory. Highest or base level directory in a hierarchical file system. Subdirectories branch off of the root directory.

router. An attaching device that connects two LAN segments, which use similar or different architectures, at the Reference Model network layer. Contrast with bridge and gateway.

S

scanner. A device that examines the bar code on merchandise tickets, credit cards, and employee badges and generates analog or digital signals corresponding to the bar code.

scroll. To move all or part of the display image vertically or horizontally to display data that cannot be observed within a single display image. See also page (2).

SDLC. Synchronous Data Link Control.

session. (1) A connection between two application programs that allows them to communicate. (2) In SNA, a logical connection between two network addressable units that can be activated, tailored to provide various protocols, and deactivated as requested. (3) The data transport connection resulting from a call or link between two devices. (4) The period of time during which a user of a node can communicate with an interactive system, usually the elapsed time between log on and log off. (5) In network architecture, an association of facilities necessary for establishing, maintaining, and releasing connections for communication between stations.

sign-on. (1) A procedure to be followed at a terminal or workstation to establish a link to a computer. (2) To begin a session at a workstation.

SNA. Systems Network Architecture.

socket. Synonym for port (2).

source. The origin of any data involved in a data transfer.

state. See conversation state.

subdirectory. Any level of file directory lower than the root directory within a hierarchical file system.

subroutine. Section of code that performs a specific task and is logically separate from the rest of the program.

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

supervisory (S) frame. A frame in supervisory format used to transfer supervisory control functions. See also information frame, unnumbered frame.

synchronous. (1) Pertaining to two or more processes that depend upon the occurrence of a specific event such as a common timing signal. (2) Occurring with a regular or predictable timing relationship.

Synchronous Data Link Control (SDLC). A discipline conforming to subsets of the Advanced Data Communication Control Procedures (ADCCP) of the American National Standards Institute (ANSI) and High-level Data Link Control (HDLC) of the International Organization for Standardization, for managing synchronous, code-transparent, serial-by-bit information transfer over a link connection. Transmission exchanges may be duplex or half-duplex over switched or nonswitched links. The configuration of the link connection may be point-to-point, multipoint, or loop.

system. (1) In data processing, a collection of people, machines, and methods organized to accomplish a set of specific functions. See also data processing system and operating system. (2) In the QVS Distributed Data Services, a group of nodes for which files are managed.

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

system ID. In the QVS Distributed Data Services, the identifier for a system. Each system ID must be a unique value for each system in environments where multiple systems are interconnected via bridges. System IDs are specified at each node during the installation of the QVS Distributed Data Services. A system ID cannot be changed without reinstalling the QVS Distributed Data Services.

Systems Network Architecture (SNA). The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through, and controlling the configuration and operation of, networks.

Note: The layered structure of SNA allows the ultimate origins and destinations of information, that is, the end users, to be independent of, and unaffected by, the specific SNA network services and facilities used for information exchange.

T

task. A basic unit of work.

terminal. In data communication, a device, usually equipped with a keyboard and a display, capable of sending and receiving information over a communication channel.

threshold. (1) A level, point, or value above which something is true or will take place and below which it is not true or will not take place. (2) In IBM bridge programs, a value set for the maximum number of frames that are not forwarded across a bridge due to errors, before a “threshold exceeded” occurrence is counted and indicated to network management programs. (3) An initial value from which a counter is decremented from an initial value. When the counter reaches zero or the threshold value, a decision is made and/or an event occurs.

token. A sequence of bits passed from one device to another on the token-ring network that signifies permission to transmit over the network. It consists of a starting delimiter, an access control field, and an end delimiter. The frame control field contains a token bit that indicates to a receiving device that the token is ready to accept information. If a device has data to send along the network, it appends the data to the token. When data is appended, the token then becomes a frame. See frame.

token ring. A network with a ring topology that passes tokens from one attaching device (node) to another. A node that is ready to send can capture a token and insert data for transmission.

trace. (1) A record of the execution of a computer program. It exhibits the sequences in which the instructions were executed. (2) A record of the frames and bytes transmitted on a network.

transmission. The sending of data from one place for reception elsewhere.

transmission frame. See frame.

U

underflow exception. A condition caused by the result of an arithmetic operation having a magnitude less than the smallest possible nonzero number. See also overflow exception.

unnumbered (U) frame. A frame in unnumbered format, used to transfer unnumbered control functions. See also information frame, supervisory frame.

user. (1) Category of identification defined for file access protection. (2) A person using a program or system.

user interface. Hardware, software, or both that allows a user to interact with and perform operations on a system, program, or device.

V

variable. (1) A named entity that is used to refer to data and to which values can be assigned. Its attributes remain constant, but it can refer to different values at different times. (2) In computer programming, a character or group of characters that refers to a value and, in the execution of a computer program, corresponds to an address. (3) A quantity that can assume any of a given set of values.

version. A separate IBM-licensed program, based on an existing IBM-licensed program, that usually has significant new code or new function.

W

work area. An area in which terminal devices (such as displays, keyboards, and printers) are located. Access units may also be located in work areas.

workstation. (1) An I/O device that allows either transmission of data or the reception of data (or both) from a host system, as needed to perform a job: for example, a display station or printer. (2) A configuration of I/O equipment at which an operator works. (3) A terminal or microcomputer, usually one connected to a mainframe or network, at which a user can perform tasks. (4) In the QVS Distributed Data Services, a LAN-attached machine that is running the QVS Distributed Data Services. Synonym for node.

