UNISYS

Product Information Announcement

o New Release ● Revision o Update o New Mail Code

Title

DCP Series Distributed Communications Processor Operating System (DCP/OS) Operations Reference Manual, Level 5R3

This Product Information Announcement announces the new release of the DCP Series Distributed Communications Processor Operating System (DCP/OS) Operations Reference Manual, Level 5R3 (7831 5702).

The Distributed Communications Processor Operating System (DCP/OS) supports a multi-user environment for building, loading, and executing programs. The operating system is part of a total communications environment that uses Distributed Communications Processors (single and multi) to implement intelligent network applications.

The DCP/OS Level 5R3 Operations Reference Manual provides information necessary to operate DCP/OS, Level 5R3 in a communications network environment. In particular, this manual describes the following:

- Booting and dumping the DCP/OS
- Entering and using DCP/OS commands
- System tuning and memory management
- Using utility programs
- Identifying error codes

This new release contains information associated with DCP/OS Level 5R3 as follows:

- Enhancements and additions to the Console Mode Commands
- Enhancements and additions to the Demand Mode Commands

To receive the complete manual for this new release, order 7831 5702–100.
To order additional copies of this document

- United States customers call Unisys Direct at 1-800-448-1424
- All other customers contact your Unisys Subsidiary Librarian
- Unisys personnel use the Electronic Literature Ordering (ELO) system

Announcement only	Announcement and attachments	System: DCP Series
MU59, MAC, MBZ, MHA, MUY1, MX3, MX5,MX6, MX8, MY1, MY3, MY5, MY6, MY7, MUS1, MUR3	AF01	Release: 5R3 Part number: 7831 5702–100

UNISYS

DCP Series
Distributed Communications Processor Operating System (DCP/OS)
Operations Reference Manual

Release Level 5R3

March 1993

Printed in U S America
7831 5702–100

Priced Item

Printed on recycled paper

7831 5702-100

Page Status

Page	Issue
iii	–000
iv	Blank
v through xxii	–000
1–1 through 1–7	–000
1–8	Blank
2–1 through 2–26	–000
3–1 through 3–46	–000
4–1 through 4–135	–000
4–136	Blank
5–1 through 5–17	–000
5–18	Blank
6–1 through 6–11	–000
6–12	Blank
7–1 through 7–5	–000
7–6	Blank
A–1 through A–17	–000
A–18	Blank
B–1 through B–26	–000
C–1 through C–2	–000
D–1 through D–18	–000
E–1 through E–5	–000
E–6	Blank
Glossary–1 through 10	–000
Bibliography–1 through 2	–000
Index–1 through 11	–000

About This Manual

Audience Description

This manual is intended for Distibuted Communications Processor Operating System (DCP/OS) programmers and operators. It is a reference and a guide for performing specific operations and control job tasks, and interpreting messages and system error codes.

Purpose

This manual is part of the operations subset of the Communications Delivery library whose purpose is to familiarize the user with Distributed Communications Processor Operating System (DCP/OS) procedures and commands.

The DCP/OS supports a multi-user/multi-tasking/multi-processing environment for building, loading, and executing programs. The operating system is part of a total communications environment that uses DCPs to implement intelligent network applications.

The communications network has three major software components:

- Communications Management System (CMS 1100)
- Telcon
- Distributed Communications Processor Operating System (DCP/OS)

This manual covers the DCP/OS functions.

Scope

This manual describes the following basic DCP/OS operations:

- Booting and dumping the DCP
- Entering and using DCP/OS commands and utilities
- System tuning and memory management
- Executing utility programs
- Interpreting messages
- Identifying error codes

Prerequisites

This manual assumes basic computer and communications knowledge.

How to Use This Manual

This manual is a reference which contains an enormous amount of technical information and requires careful study. If not familiar with the Distributed Communications Processor Operating System (DCP/OS), the user is advised to first read the glossary to become acquainted with the terminology.

Organization

This manual is divided into the following sections and appendices.

Section 1. Introduction

This section describes the DCP operating system. The overview discusses the DCP/OS functions and illustrates the communications environment. The various utility programs, command modes, workstations, and on-line help are also described.

Section 2. System Booting and Dumping

This section tells how to boot the DCP and set the various switches. It also discusses the contents of the boot diskette and explains the loading process. This section includes illustrations of the operator control panels and the maintenance panels.

Section 3. Console Mode Commands

This section describes the console mode and the proper use of command syntax. It also provides the function and format of each console mode command and explains how to send output to a printer.

Section 4. Demand Mode Commands

This section describes demand mode commands and a number of utility programs. The function and format of each command and utility program is provided.

Section 5. Demand Mode Bypass Commands

This section explains the use of demand mode bypass commands and how to enter these commands. It also provides the function and format of each command.

Section 6. Memory Management

This section describes how DCP memory works and explains disk caching, thresholds and throttling, resident and transient segments, and system tuning.

Section 7. Support

This section explains how to report software and documentation problems.

Appendix A. System Error Codes

This appendix lists the error codes and their descriptions.

Appendix B. OS 1100-Based Utilities

This appendix covers OS 1100-based utilities. It explains how to convert OS 1100 absolute elements to DCP-formatted elements, how to analyze DCP dumps, how to convert OS 1100 files, and how to transfer OS 1100 files to or from a front-end DCP.

Appendix C. Non-Supported Utilities

This appendix lists some useful but not critical utilities in the **DCPLIBABS.**file on the DCP/OS release tape such as @BACKUP — a manager for securing files, and @FCOMP — a utility for comparing files or elements.

Appendix D. Workstation Messages

This appendix lists the workstation messages that may appear on the console, explains the messages, and gives examples for recovery where appropriate.

Appendix E. Common Line Module Identifiers

This appendix lists common line module identifiers.

Related Product Information

Manuals are referenced in text using a shortened version of the title. To make it easy for you to find them here, they are listed alphabetically by the shortened title followed by the full title. An annotated list of documents that are directly related to the subject of this document follows.

CMS 1100 Operations Reference Manual
Full title:
OS 1100 Communications Management System (CMS 1100) Operations Reference Manual (7831 5694). Previous document number: UP-9689.

This manual is part of the operations subset of the Communications Delivery library. The communications network has three major software components: Communications Management System (CMS 1100), Telcon, and Distributed Communications Processor Operating System (DCP/OS). This manual provides operations information for the CMS software.

Communications Delivery Configuration Guide
Full title:
OS 1100/DCP Series Communications Delivery Software Configuration Guide (7831 5678). Previous document number: UP-9957.

This guide tells how to configure Communications Delivery software for a data communications network. It also tells you how to reconfigure these software products as your network evolves.

Communications Delivery Configuration Reference Manual
Full title:
OS 1100/DCP Series Communications Delivery Software Configuration Reference Manual (7831 5686). Previous document number: UP-11580.

This manual provides reference material for configuring data communications networks with Communications Delivery software.

Communications Delivery Installation Guide
Full title:
OS 1100/DCP Series Communications Delivery Software Installation Guide (7831 5645). Previous document number: UP-9956.

This guide tells you how to generate, install, and verify Communications Delivery software on an OS 1100 host and its Distributed Communications Processors (DCPs). Generating and installing involves copying the Communications Delivery software components and related software products from release tapes to mass storage and preparing the software for use with your communications network.

DCA CPA Programming Reference Manual
Full title:
OS 1100 Distributed Communications Architecture (DCA) Technical Overview (7431 5805)

This manual is for potential owners, managers, and users of distributed data processing systems structured according to DCA Level II.

DCP/OS Programming Reference Manual
Full title:
DCP Series Distributed Communications Processor Operating System (DCP/OS) Programming Reference Manual (UP-11540).

This manual contains information for writing programs to run under the DCP Operating System (DCP/OS) on a Distributed Communications Processor (DCP).

Telcon Operations Guide
Full title:
DCP Series Telcon Operations Guide (7831 5785). Previous document number: UP-13431.

This manual is part of the operations subset of the Communications Delivery library. It is a guide to Telcon operations. It explains how a DCP network is organized, to use Network Management Services (NMS) consoles and commands, to use Telcon online configuration, to transfer files in a DCP network environment, to turn on instrumentation, to interpret messages, and to control console and logged messages.

Telcon Operations Reference Manual
Full title:
DCP Series Telcon Operations Reference Manual (7831 5728). Previous document number: UP-9256.

This manual is part of the operations subset of the Communications Delivery library. It is a reference to the full range of options on NMS commands and online configuration commands. It lists online hardware verification operations, Remote File System (RFS) commands used to transfer files, hardware instrumentation parameters on the TRON command, general NMS console messages, and CENL console messages.

Throughout this manual, when you are referred to another manual, use the version that applies to the software level in use at your site.

Unless otherwise noted, the term DCP refers to all DCP models including DCP/5,15,25,30,35,50,55,612, 614, 618, 622, and 624.

Command Notation Conventions

This manual uses the conventions that follow to present command formats. It distinguishes between notation conventions, symbols, and required characters in syntax.

COMMAND *parameter-string*

where:

COMMAND	is the name of a command. Command names are presented in SMALLCAP letters to indicate that you must enter them exactly as shown.

parameter-string	is the parameter string for the command. These parameter strings are lowercase letters or italic when the exact input is variable.

- Brackets ([ ]) indicate optional information.
- Braces ({ }) indicate that you must choose one of the items shown.
- Braces within brackets ([{ }]) indicate that you may choose only one of the optional items.
- Names enclosed in angle brackets (< >) describe a class of symbols. These are sometimes called nonterminals. Names not enclosed in angle brackets are called terminal symbols and make up numbers of a class.
- The symbol (::=) means "is defined as".
- An ellipsis (...) means the preceding items can be repeated.
- Lowercase *italic* character strings indicate names you must provide.

Key caps and other product nomenclature	All markings on key caps, controls, dials, switches, and so on, are shown in SMALLCAPS and spelled exactly as they appear on the equipment. Keys that are to be pressed in sequence are shown separated by hyphens.

Required Command Characters

You may need to use the following characters when entering a CMS 1100, Telcon, or DCP/OS command.

< >	Angle brackets are required in some cases for Telcon. Do not use them for any other purpose.
::	Double colons separate some Telcon commands.
;	Semicolons act as a continuation symbol when you continue a command on the next line. CMS 1100 commands almost always require a space before the semicolon; Telcon commands do not.
Spaces	Spaces in commands represent the actual number of spaces you must enter as part of a command.

Additional command notation conventions are given in the following sections:

- Section 3, "Console Mode Commands"
- Section 4, "Demand Mode Commands"
- Section 5, "Demand Mode Bypass Commands"

Contents

Figures

Tables

Section 1
Introduction

This section does the following:

- Gives an overview of the DCP/OS
- Discusses command modes
- Describes workstations
- Describes how to use online HELP

1.1. DCP/OS Overview

The Distributed Communications Processor Operating System (DCP/OS) supports a multi-user/multi-tasking/multi-processing environment for building, loading, and executing programs on Distributed Communications Processors (DCPs). The DCP/OS acts as a resource manager. It handles most of the memory management, file control, and service utilities required for basic operation. It can run network and gateway packages as independent applications.

The DCP/OS also handles Communications Processor Architecture (CPA) services. The CPA consists of a set of hardware-based and microcode-supported structures such as segments, procedures, and queues. See the *DCP Series Communications Processor Architecture Programming Reference Manual* (7431 5805). The main CPA functions are to maintain communications among all processing elements and to provide an efficient basis for distributed processing within the system. The architectural microcode, along with other system data, is loaded from mass storage when the system is initialized and resides in microcode local storage.

The operating system is part of a total communications environment that uses DCPs to implement intelligent network applications. This communications environment consists of several entities, shown in Figure 1–1. Entities specific to the DCP/OS are discussed in this section.

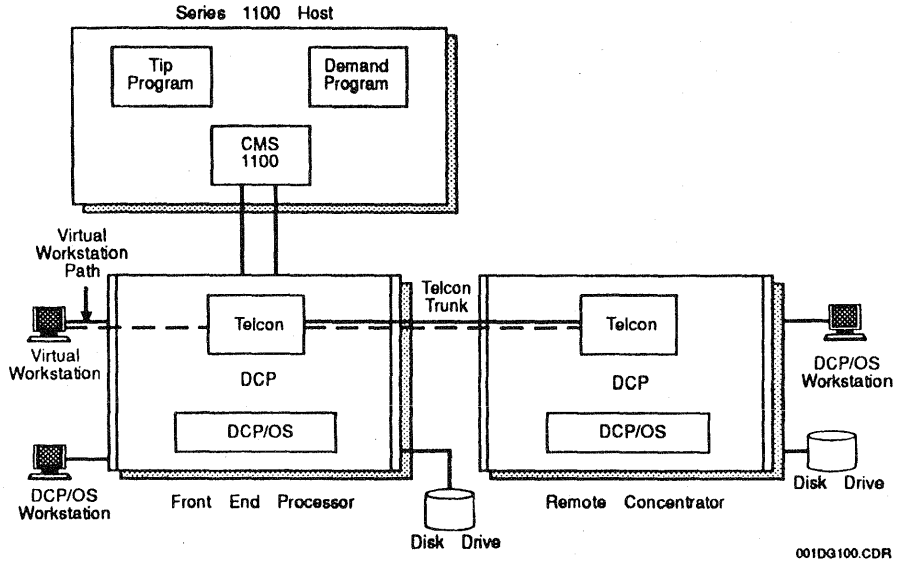


Figure 1–1. Communications Environment

The DCP/OS package includes the operating system itself and a set of system utilities. The DCP/OS executes programs that are initially downloaded to the DCP from an OS 1100 computer, a UNIX system, or installed from diskettes. For more information on the downloading procedure, see Section 2 of this manual and the *Communications Delivery Installation Guide* (7831 5645).

1.2. Utility Programs

In addition to the operating system, the DCP software package provides a number of utility programs (see 4.2). They include file manipulation utilities such as file/element copy, a DCP/OS configurator, a program builder, and a disk utility program. These programs may be run in a concurrently processing (multiprogramming) environment.

The DCP/OS package also provides a number of nonsupported utilities, which are listed in Appendix C.

1.3. Command Modes

The following operating modes are available on the DCP/OS workstation:

- Console mode
- Demand mode
- Batch mode
- Demand mode bypass
- Debug mode

Console mode is used to control the system. When the DCP/OS is initially booted, all workstations connected to ports owned by the DCP/OS are in console mode. See Section 3.

Demand mode provides an environment in which to develop and execute programs. Demand mode is started by entering the @RUN command either from a DCP/OS workstation in console mode, or from a Telcon supported workstation that has access to the target DCP/OS. You can enter further commands to invoke programs. You can direct the system to temporarily take commands from mass storage files (i.e., @ADD commands. See Section 4 for more information.

A **batch mode** run is started by the ST command in console mode or by the @START command in either demand or batch mode. In both cases, the runstream is specified by `qualifier*file.elementname` and any output is sent to a dynamically created spool file.

The spool file name is generated from a `run-name` as follows:

```
SYS$*P$nnnnn
```

nnnnn = `run-name`

Note: *The spool files are not secured in any way. If the spool file already exists when a batch job is started, it is deleted and recataloged. The spool file can be examined using the line editor @ED or displayed using @PRT.*

Demand mode bypass commands can be used only in demand mode. The processing you request by entering a demand mode bypass command runs independently of any current program execution. See Section 5.

Debug mode may be entered from console or demand mode directly or after a demand run program hits a trap. The DEBUG mode commands correct programs being developed, and restart them if necessary. For more information on debug mode commands, see the *DCP/OS Programming Reference Manual* (7431 6894).

1.4. Workstations

A workstation is either a Universal Terminal System (UTS) or a UTS emulated terminal physically connected to a port owned by the DCP/OS. In the case of the DCP/5, a workstation can be a personal computer with a server/keyboard combination. The screen operates in simple scroll mode. For more information on the DCP/5, refer to the *DCP/5 Installation and Operations Reference Manual* (UP–14133).

Workstations are not supported on multiline line modules. An operator uses a workstation to enter commands to the operating system. These commands typically display the current status or start programs. See Table 3–1 for a list of console mode commands.

Once a program is started, all workstation input is sent to the program. If the program does not have a read outstanding, the input is discarded and a wait message is returned to you. You can break in by pressing the MESSAGE WAIT key. You can then terminate the program or enter debug mode.

. Online Help

Extensive online help is provided by the DCP/OS utilities through two methods:

- The interactive @HELP program
- The H option within most @ commands.

All DCP/OS utility programs recognize the H option on the call line and respond with a help page that briefly describes the program function and its use.

Format

```
@command,H
```

Parameter

H (requests the help page for the given command)

Example

```
@MSG,H
```

Response

```
@msg,h
```

Help information:

```
@MSG,[chlw] ['](message text)[']
```

Explanation

options:

C	means send message to operator
H	means print help page
L	means display messages (for example, banners)
W	means wait for input after print; does not wait in batch mode; ignored if using C option.
(message text)	is the string to be printed. Quotation marks are required only if the message string contains blanks.

Section 2
System Booting and Dumping

This section does the following:

- Describes the contents of the boot diskettes
- Gives the steps for copying a DCP/5 boot diskette
- Explains how to boot the DCP/OS and monitor boot status
- Explains how to dump DCP memory and monitor dump status
- Describes and illustrates DCP control panels

2.1. Contents of the Boot Diskettes

The boot diskettes delivered with this Communications Delivery are one of the following types:

- 3 ½-inch (DCP/5,25,30,35,50,55,600 Series)
- 5 ¼-inch (DCP/5,15)

The 3 ½-inch and 5 ¼-inch diskettes contain the DCP microcode and bootstrap software as well. In addition to the DCP microcode and bootstrap software, the 3 ½-inch and 5 ¼-inch diskettes contain the file **SYS$*SYSMONX.** and extra utilities in **SYS$*SYSLIBX.** This extra capacity allows a system to be booted from the diskette.

The DCP/5 requires two diskettes: one for installation and one for microcode. The diskettes can be either 3 ½-inch or 5 ¼-inch.

Table 2–1 describes the contents of the boot diskettes.

Table 2–1. Boot Diskette Files for 5-¼ and 3 ½-inch Diskettes

File	5 ¼ and 3 ½-inch
SYS$*SYSBOOT (&SYS$*SYSOVL)	Bootstrap
SYS$*SYSCFG	DCP/OS configuration
SYS$*SYSLMCX *	Boot-time LM programs
SYS$*SYSJOBX *	Boot-time runstreams
SYS$*SYSLIBX *	Program library
SYS$*SYSMONX	DCP/OS

* = FILE.ELT FORMATTED FILES

Notes:

1. *All boot diskette files can be copied by using the @COPY command.*
2. *You cannot copy elements to diskettes with 512 bytes/sector format.*
3. *You can copy elements to diskettes with 256 bytes/sector format.*
4. *The 3 ½-inch diskette and 5 ¼-inch diskette program libraries contain DOWNLOAD, FUP, and MONFIG.*

2.2. Duplicating a DCP/5 Boot Diskette

There are certain idiosyncracies that make the process of duplicating a boot diskette for a DCP/5 different from that for other DCPs. The recommended procedure is as follows:

1. Copy the boot diskette image into a file on the DCP/5 (let's assume that the file has been given the name **Q*BOOTIMAG.**). This image can be copied from the current boot diskette (inserted onto device SD0) with the following command:

   ```
   @@CONS UP SD0
   @@CONS DN SD0
   @DKCOPY SD0,Q*BOOTIMAG.
   ```

   The @DKCOPY utility catalogs the **Q*BOOTIMAG** file automatically.

   Occasionally, disk images for new bootstrap levels may be copied from a Unisys host. For information on where to find the correct disk image and how to move it to the DCP/5, contact your customer support representative.

2. Format the new boot diskette at 720 Kbytes. On some PCs, this can be done with the following DOS command:

   ```
   FORMAT A: /F:720
   ```

   However, on many PCs it is necessary to use public domain PC program FDFORMAT, which is available from your customer support representative. The syntax is:

   ```
   FDFORMAT A: H2 S9 T80
   ```

   Alternatively, the diskette may be formatted on a different DCP (not a DCP/5) using the DCP/OS utility @DISK. The diskette should be formatted at 512 bytes per sector.

3. With the new diskette in SD0, purge it using the DCP/OS utility @DISK. This initializes the control structures on the diskette (the Volume Space Table, Data Set Labels, etc.).

4. Use the UP and DN commands on the diskette to allow DCP/OS to recognize its density and update internal tables:

   ```
   @@CONS UP SD0
   @@CONS dn SD0
   ```

5. With the boot diskette image in the DCP file **Q*BOOTIMAG.**, copy it to the new diskette with the @DKCOPY utility

```
@DKCOPY,V Q*BOOTIMAG.,SD0
```

2.3. Booting the System

There are five types of boot operations:

- Port path load from the OS 1100 host
- Port path load from disk
- Boot from the integrated diskette
- Load from disk using configured load paths
- Load from host using configured load paths

Tell the DCP which type of boot to perform by using the LOAD switches on the DCP. The basic LOAD settings are described in Table 2–2.

Table 2–2. Basic Load Settings

Setting	Operation
All switches down (zeroes).	Load from the integrated diskette (typically for troubleshooting or when a host is unavailable).
Leftmost switch up and port number in other switches.	Load using designated port.
Leftmost switch down and load path in other switches.	Load using load path.

You can perform any of the boot operations by following this sequence:

1. Turn on the equipment.
2. Insert the system diskette. The label should face either up, down, or to the right, depending on the DCP model.
3. Set the switches according to DCP type, as specified in Table 2–3.

Note: *The position of any unspecified switch does not affect the boot process.*

4. Set the LOAD switches. The number of LOAD switches varies according to the DCP type. In all cases, however, the switches represent binary number values where the DOWN position indicates a 0 value and UP indicates a 1. Set these switches as shown in Table 2-4.

5. Push either the SYSTEM RESET button or the PROGRAM LOAD button. Table 2–5 explains the differences between these buttons.

Notes:

1. *The DCP/5 uses the same load switch philosophy as the other DCP models, but implements the switches through screen menu selections. For more information on how to set the load switches on the DCP/5, refer to the DCP/5 Installation and Operations Reference Manual (UP–14133).*
2. *The DCP612,614,618,622, and 624 use the same load switch philosophy as the other DCP models, but implements the switches through a window on the Control Station.*

Table 2–3. Normal Switch Positions for Booting

DCP Type	Normal Switch Settings
DCP/5	Refer to the *DCP/5 Installation and Operations Reference Manual* (UP–14133).
DCP/15	Set the LOCAL/REMOTE switch to the LOCAL (down) position.
DCP/25,30,35	Set the MAINTENANCE switch to NORMAL (down).
DCP/50,55	Set the MAINTENANCE switch to NORMAL (down).
DCP/612,614,618,622,624	

Table 2–4. Load Switch Positions

Load Type	Switches	Notes
IFD	All switches DOWN (zeros)	IFD = integrated flexible diskette (the boot diskette)
Port Path	Leftmost switch UP. All other switches set to port path.	
Load path	Leftmost switch DOWN. All others switches set to load path (one or more must be UP).	

Table 2–5. SYSTEM RESET and PROGRAM LOAD Functions

Button	Functions
SYSTEM RESET	0. Clear all line modules (LMs) 1. Sets memory to zeroes. 2. Loads DCP microcode from boot diskette. 3. Loads the DCP/OS's bootstrap from boot diskette. 4. Loads the DCP/OS
PROGRAM LOAD	0. Clear all line modules (LMs) 1. Loads partial DCP microcode from boot diskette. 2. Loads DCP/OS's bootstrap from boot diskette. 3. Dumps DCP memory and loads DCP/OS.

Push SYSTEM RESET for a boot to clear memory. Push PROGRAM LOAD for a dump of DCP memory. You can follow the progress of the load operation by watching the display lights on the DCP and by watching the system console.

Mode clear is issued by the partition's IOPs to the LMs on both system reset and program load (if that partition is of a DCP/25,30,35,50,55,612,614,618,622, and 624 machines, and on program load for DCP/5,15 machines). The hardware clear is issued by the hardware to the LMs on system reset. This results in the following states (Table 2–6) for the various LM categories:

Table 2–6. LM Catagory States

If the LM was still	Then the DBILM (on program load) (on system reset) is... 1	The DBLM (on program load) (on system reset) is... 1	The SBLM (on program load) is... 2	And the SBLM (on system reset) is... 2
operational	still operational	not operational	not operational	not operational
loaded	still loaded	still loaded	still loaded	not loaded
connected to partition's bus	not connected	not connected	not connected	not connected

Notes:

1. *If the LM was not connected to the partition's bus at the time of the system reset or program load, the LM is not affected at all.*
2. *If the LM is not associated with the partition, the LM is not affected at all*

For an initial download of the DCP from a host, you should determine the port number of the host channel or communications line. Enter this number into the LOAD switches and set the leftmost (0 position) switch to UP. You do not need to determine the port number if you created DOWNLOAD load path with @MONFIG.

Note: *When you have completed booting the system, set the switches to the dump path that you want to use.*

Caution

To dump DCP memory, press PROGRAM LOAD. Do not press SYSTEM RESET; this will clear memory.

2.3.1. Example: Booting a DCP/50

This example explains how to boot a DCP/50 by downloading the software from an OS 1100 host. Assume that the host channel is on port number 0E in the DCP/50.

1. Before downloading, prepare the host for the load. See the *Communications Delivery Installation Guide* (7831 5645) for details on preparing the download file and configuring CMS 1100.
2. Ensure that the DCP/50 is powered on.
3. Insert the 3 ½-inch system diskette.
4. Set the MAINTENANCE switch to the NORMAL position (down).
5. Set the LOAD switches to 1000 1110 (port path, using port 0E).
6. Push the SYSTEM RESET button.

Note: *For more information on the DCP file transfer program, refer to Appendix B.*

2.3.2. Booting a Partition on a DCP600 Series

The DCP600 has either one or two partitions. The diskette for the top IOM (partition A) is labeled 'partition A', while the diskette for the middle IOM (partition B) is labeled 'partition B'.

This example explains how to boot a partition on a DCP600 series by downloading the software from an OS 1100 host. Assume that the host channel is on port number 0E in the DCP600.

1. Set the power switch at the rear of the control station to ON. The DCP Control window is displayed as shown below in Figure 2–1.

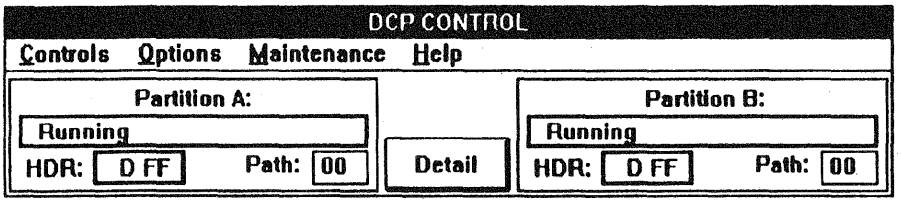


Figure 2–1. DCP Control Window

Note: *The word 'click' is used on this procedure as a replacement for 'position the mouse pointer on a menu item and press the left mouse button".*

2. Click on the Controls menu on the menu bar of the DCP Control window.
3. Click on the Partition A menu item as shown in Figure 2–2.

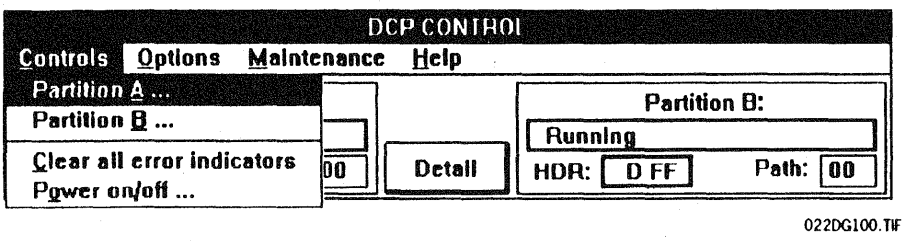


Figure 2–2. Controls Menu

4. Set the hexidecimal code of 8E into the load switched by clicking on the zero (0) buttons in the Partition A - Controls window (as shown below in Figure 2–3).

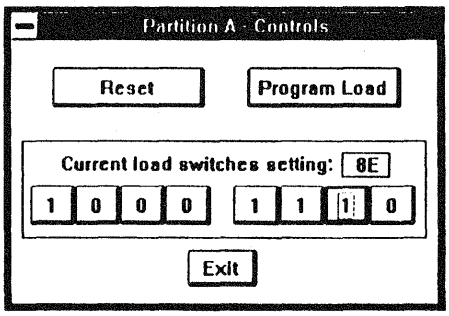


Figure 2–3. Partition A - Controls Window

Observe the new load switch value is now 8E, then click on the Exit button.

5. Click on the Controls menu on the menu bar of the DCP Control window.
6. Click on the Partition A menu item to see the Partition A — Controls window.
7. Click on the Reset menu button. A Warning dialog box appears on the screen (Figure 2–4).

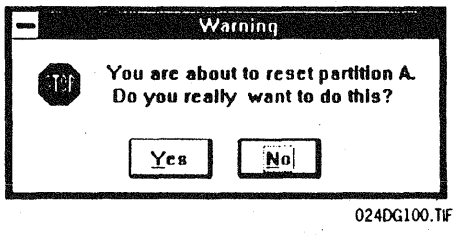


Figure 2–4. Reset Warning Dialog Box

8. Click on the Yes button of the dialog box. The DCP Control window is displayed again.

Watch the HDR (hardware display register) line and wait for the code DFF to appear. Code DFF indicates the DCP600 has completed the OS software load and is in an idle loop.

Observe that the DCP/OS console is now polling and the OS banner is displayed on the screen of the DCP/OS console.

3. Monitoring Boot Status

Each DCP has a hardware display (either a two- or three-digit hexadecimal LED DISPLAY or eight STATUS lights) that indicates the load status of the system.

Note: *Status codes are displayed within a status window on the DCP/5,612,614,618,622, and 624.*

Table 2–7 describes boot status codes.

Table 2–7. Boot Status Codes

Description	Hexadecimal Code	Binary Code
Boot started	(B)B0	10110000
Alternate load path taken	(B)B1	10110001
Successor load path taken	(B)B2	10110010
Waiting for host (download only)	(B)B3	10110011
Copying DCP/OS from host (download only)	(B)B4	10110100
Loading DCP/OS	(B)B5	10110101
Multiple processors active	(B)B9	10111001
Boot complete	(B)BC	10111100

Refer to Table 2–14 for dump status codes.

After the system has finished booting the DCP/OS, the display lights normally contain one of the following hexadecimal values:

- (D)FF System is idle, initializing, or switching between runs and idle.

- (D)00 DCP/OS task is active.
- (D)xx User task is active for run xx.

In case of a system error, the display lights indicate the error status. On those DCP types with the two- or three-hexadecimal digit display, the display alternately shows the upper byte and the lower byte of the 2-byte (16-bit) error code. On the DCPs with STATUS lights, the LOAD switches must be set in the DOWN position to avoid corrupting the error status display.

Appendix A contains descriptions of the system error codes.

2.3.4. Startup Runstreams

If you are an experienced operator and knowledgeable about the startup runstreams, you may want to alter them to accommodate special circumstances at your site. The startup runstream STARTDOWN is found on the boot diskette in the file **SYS$*SYSJOBX** (refer to Table 2–11 for additional runstreams). Put any changes you make on the INSTALL runstream on the host so you can track changes.

For local loads, the runstream element is **SYS$*SYSJOB.STARTUP**. For host loads, it is **SYS$*SYSJOBX.STARTDOWN**; however, the download process also eventually starts the PRE-INSTALL, INSTALL, POST-INSTALL, and finally the STARTUP runstreams (all usually located in the file **SYS$*SYSJOB**).

The PRE-INSTALL and POST-INSTALL runstreams are automatically generated by the DOWNLOAD program, which is started by the STARTDOWN runstream. INSTALL and STARTUP are generally downloaded from the host.

Tables 2–8 and 2–9 explain the contents of these runstreams. Table 2–8 describes the contents of local load runstream.

Table 2–8. Local Loads

Runstream	Contents	Comments
STARTUP	Variable	Found in **SYS$*SYSJOB**

Table 2–9 describes the contents of remote load runstreams.

Table 2–9. Remote Loads

Runstream	Contents	Comments
STARTDOWN		Found in **SYS$*SYSJOBX** on the boot diskette
	DOWNLOAD	Program found on the boot diskette in **SYS$*SYSLIBX**
	PRE-INSTALL	Runstream generated by DOWNLOAD program
	INSTALL	Runstream downloaded from host
	POST-INSTALL	Runstream generated by DOWNLOAD program
	STARTUP	Runstream downloaded from host

Note: *Use remote loads only for initial loads and other special circumstances. Local loads are preferred unless you are updating to a new software level.*

2.4. Dumping the System

There are two types of dumps: system or machine dumps and program dumps.

System or machine dumps:

- Dump all of memory in its state at the time of the dump
- Do not always contain progam or DCP/OS dictionaries
- Are followed by rebooting the DCP
- Are caused by fatal system errors or operator intervention (PROGRAM LOAD)

There are two types of system or machine dump operations:

- Local (dumping to a local disk file) through a load path setting
- Remote (dumping to a remote host) through either a port or load path setting (UPDUMP)

Program dumps:

- Dump programs and DCP/OS memory only, under normal @CRASH options (P option)
- Always include the dictionary of the program being dumped
- Can terminate the program and do not cause a reboot
- Are caused by program error, SNAP, or the Debug D command (only program error terminates the program)

Dumping the system means dumping all of the DCP memory. This includes the DCP/OS itself and any active batch and demand programs. Dumping of individual programs operating under the DCP/OS is discussed in 4.2.8, @CRASH. For more information on dumping Telcon runs, see the *Telcon Operations Reference Manual* (7831 5728).

You might need to take a complete memory dump for one of the following reasons concerning the DCP/OS:

- DCP/OS system program failure
- System hang

In the case of a fatal system error, the DCP may automatically induce a program load. When this occurs, the DCP bootstrap will read the current load switches and perform the function indicated by their value. You need to make sure that your load switches are set correctly.

Note: *To override the switches (program load only), see the @BOOTER utility in Appendix C.*

Once you have completed booting the system, set the switches to the dump path that you want to use. If you take a dump by pressing PROGRAM LOAD, you can always take the time to reset the switches beforehand. But if the program load is in progress, it is too late to change those switches. Ideally, the switches should be set to take a dump. The following switches trigger the defined actions:

Switches	**Action**
0..0	Load system from integrated flexible diskette (IFD). (The DCP cannot be dumped to the IFD.)
1nn..n	Dump system to port nn..n and reload DCP/OS over the same port (port nn..n must be a host channel or communications line).
0nn..n	Dump and/or load according to the configured load path nn..n.

Note: *Programs that fail can be dumped without rebooting the DCP. See the description of the @CRASH utility. See also the SNAP command.*

.1. Dumping to Disk (Local)

The major advantages of local dumps to disk are that they are faster to perform and can be performed even when the host channel is not working. Depending on the situation, you may not have to transfer the dump to the host for processing. You can look at it using the @DMPI or @IDUMP commands (see Section 4).

To dump the DCP memory to disk, use the @DOWNLOAD command with the F option at system initialization. This command will cause DOWNLOAD to automatically catalog a dump file of the correct size, providing there is room on mass storage, and configure load path 02 to take a local dump to a file whose size depends on the amount of memory. The dump file is initially called **SYS$*SYSDUMP.**

Note: The initial download must use the F *option or no dump file will be created.*

If you want to catalog and configure your own system dump file, use the @CAT and @MONFIG commands described in Section 4. If the dump is successful, load path 02 will automatically reload the DCP/OS from the system volume. Load path 02 chains to load path 01.

This means that it performs the dump from load path 02, then goes to load path 01 (also configured automatically by DOWNLOAD), where it is instructed to reload the operating system.

After the dump is complete, the information has been placed into a file called **SYS$*SYSDUMP** (or your own dump file). You can then retrieve this information and look at it later. If you transfer the dump to the host, you will need to retrieve this file. For more information on retrieving files, use the @COPY command in Section 4 or see the XFER command in the *Telcon Operations Reference Manual* (7831 5728).

Note: The downloading process deletes the ***SYS$*SYSDUMP*** *file and creates a new one. If the system is configured to do a local dump to that file, followed immediately by a* DOWNLOAD*, the local dump will be deleted. For more information, refer to the* @MONFIG *command in Section 4.*

2.4.2. Dumping to Host (Remote)

The major advantage of a host dump is that the dump is sent immediately to the host and can be processed there.

When the DCP is initially downloaded from a host, the downloaded DCP/OS is copied to an automatically selected system disk (designated the system volume) and Table 2–10 shows the load paths which are automatically configured:

Table 2–10. Configured Load Paths

Path	Action
1	Local load of DCP/OS from the system volume
2	Local dump of the DCP to the first disk with enough space
3	Download from host using the original port path number
4	Updump to host using the original port path number

To dump to the host, use either a load path configured for a host dump (such as load path 3), or set the load path switches to a port path to the host.

The leftmost (0 position) switch is the controlling switch in all DCP models. When the 0 position switch is up, it is in port path load mode and the operating system ignores whatever information is in the LPT. The operating system will dump according to how the physical switches are set. If the 0 position switch is down, the DCP/OS will look to the LPT for the dump path.

3. Processing the Dump on the Host

Before processing the dump on the host, create a runstream on the host. Once the runstream has been created, you can use it over and over for each subsequent dump.

The dump file on the host is in raw format. The following example runstream processes the dump into a format that you can read. In this example, the raw dump is in the file **your*dumpfile**. The file **qual*filename** contains the DCPFOR and DCPDUMP utilities (see Appendix B for further information on these utility programs).

You must modify the runstream with the options you require for your site. See Appendix B for additional options and parameters.

```
@run[,/options] run-name,account,project ID
▸@cat,p qual*printfile., ///9999
▸@asg,a qual*filename.
▸@prt,f your*dumpfile.
▸@qual*filename.dcpfor your*dumpfile.
▸@qual*filename.dcpdump new$dump.,qual*filename.dcpos-dic
▸ALL
▸@eof
```

Note: *Copy* ***qual*filename*** *from* ***DCP1100ABS*** *file on the release tape. See the release tape description in the Communications Delivery Installation Guide (7831 5645) for its location on the release tape.*

After the dump is on the host, start the runstream that you created. Verify that the dump is formatted correctly and processed completely. Finally, determine where the error occurred. If you cannot find a solution, send a User Communication Form (UCF) and a hard copy of the dump to your Unisys representative.

2.4.4. Using the Auto Reboot Key

The auto reboot key, which is normally set for each load path used to boot the DCP, allows the DCP/OS to reboot automatically after a fatal system error. Clearing the auto reboot key is done only in unusual cases for debugging purposes, since the DCP will not restart after an error occurs.

2.4.5. Example: Dumping a DCP/50 (Local)

Suppose that the DCP/50 has successfully booted from the host over port path 0E (see the example in 2.3.1). After a successful boot, set the LOAD switches as indicated:

0000 0010 = X'02'

If the DCP/OS hangs or you want a system dump for any reason, press the PROGRAM LOAD button (not the SYSTEM RESET button). The DCP will automatically perform the action defined for load path 2, which is a local dump to a file on the system volume. (The name of the dump file is initially **SYS$*SYSDUMP**.) If the dump is taken successfully, load path 2 automatically chains to 'load path 1' which reloads the DCP/OS from the system volume.

You can then use the @DMPI or @IDUMP utility on the DCP to examine the dump file (see Section 4) or you can transfer the dump file to a host for analysis. You can then use the DCPDUMP utility on the OS 1100 host.

5. Example: Dumping a DCP600 Series

For the 600 Series, the dumping procedure is slightly different from that of a DCP/50 in that there are no physical switches to set: everything is done from the DCP Control window.

1. Using the Partition A (or B) Control menu, set the load switches for a local mass storage load (dump) of a host load (dump) as shown in Table 2–11:

Table 2–11. Dump Switch Settings

If you want to perform a...	Then set load switches 0-7 to...
Local Storage Dump	00000010
Host Storage Dump	00000100

2. After the switches are set, click on the Program Load button. System memory is dumped to the specified location, and the system is reinitialized.

A Unisys representative will be able to analyze the dump using either a program residing in the host, or with DCP on-line utilities.

2.4.7. Monitoring Dump Status

As with the boot operation, you can monitor the progress of the dump by watching the display lights on the DCP and the system console.

The hardware display on the DCP indicates the status of the dump operation as described in Table 2–12.

Table 2–12. Dump Status Codes

Description	Hexadecimal Code	Binary Status
Awaiting host connection (UPDUMP only)	(B)BA	10111010
Busy dumping	(B)BB	10111011
Dump started	(B)BD	10111101
Dump ended	(B)BE	10111110

. DCP Operator Controls

Figure 2–5 illustrates the DCP/15 operator control panel.

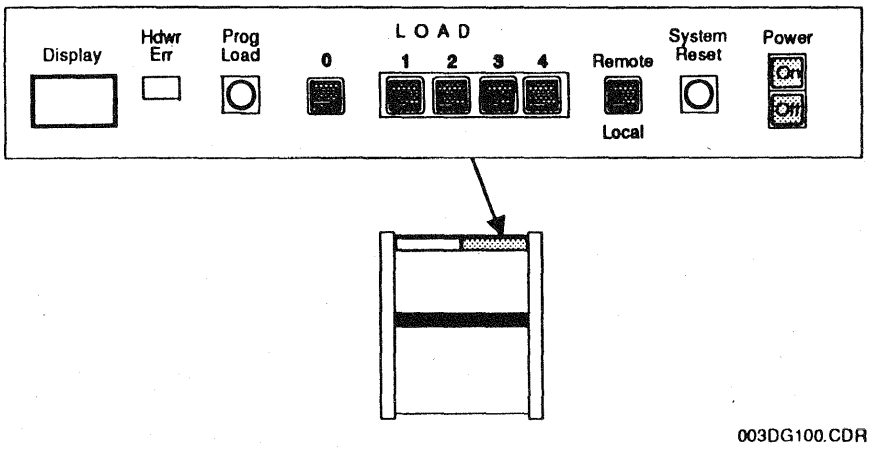


Figure 2–5. DCP/15 Operator Control Panel

Table 2–13 explains the operator controls for the DCP/15.

Table 2–13. Operator Controls for DCP/15

Control/Indicator	Function
Power ON/OFF switch	Turns power on or off and initiates a microcode load.
SYSTEM RESET pushbutton	Clears memory and boots full microcode and DCP/OS's bootstrap which loads the DCP/OS
PROG LOAD pushbutton	Boots partial microcode and DCP/OS's bootstrap which loads the DCP/OS or dumps (but does not clear) DCP memory.
REMOTE/LOCAL switch	If set to LOCAL, the switch settings on the operator panel control the actions of the DCP. If set to REMOTE, an external device (for example, another DCP) connected to the DCP through an RCM, controls the DCP.
LOAD switches	Selects load or port path number.
HDWR ERR indicator	Lights to indicate hardware failure.
DISPLAY indicator (two hexadecimal digits)	Displays code representing the current operation or status of the processor. If a DCP/OS failure occurs, the indicator alternates between the upper and lower byte of the two byte error code.

Figure 2–6 illustrates the operator control panel for the DCP/25,30,35,50,55 models.

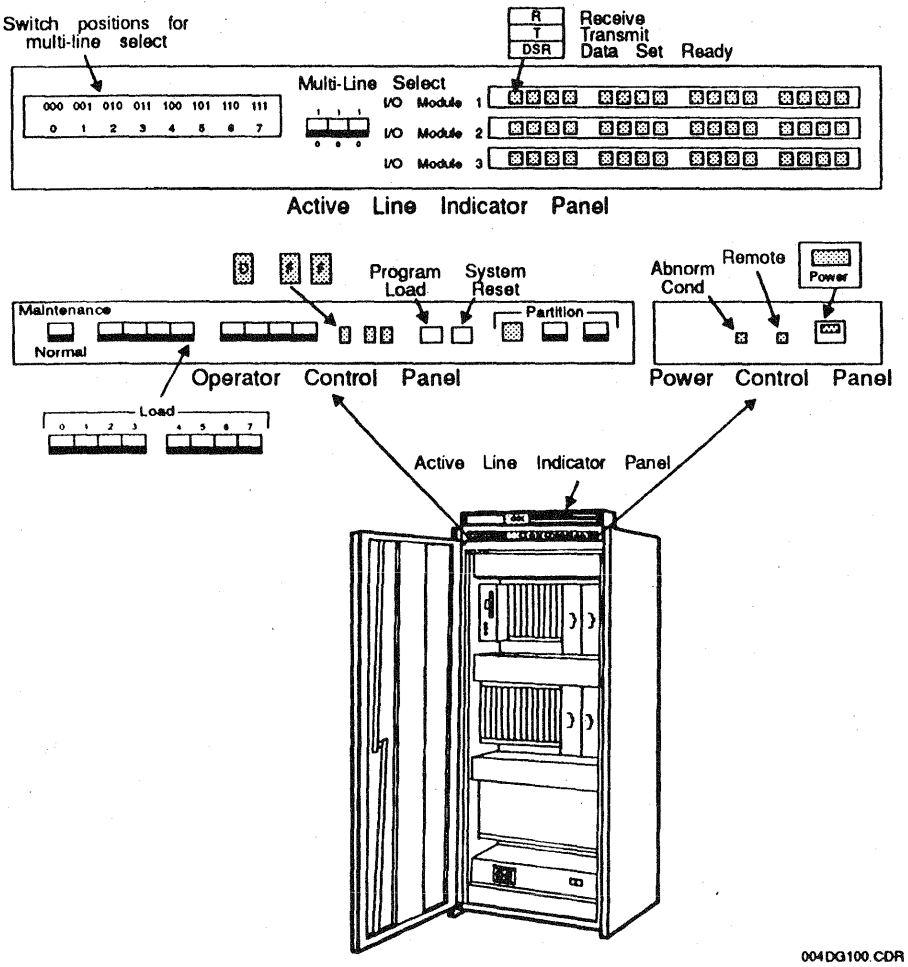


Figure 2–6. DCP/25,30,35,50,55 Operator Control Panel

Table 2–14 explains the operator controls for the DCP/25,30,35,50,55 systems.

Table 2–14. DCP/25,30,35,50,55 Operator Controls

Control/Indicator	Function
Power ON/OFF switch	Turns power on or off and initiates a microcode load.
SYSTEM RESET pushbutton	Clears memory and boots full microcode and DCP/OS.'s bootstrap which loads the DCP/OS.
LOAD switches	Selects load or port path number.
PROGRAM LOAD pushbutton	Boots partial microcode and DCP/OS's bootstrap which loads the DCP/OS or dumps DCP memory.
PARTITION switches	Must be set to OFF (00).
MAINTENANCE/NORMAL switch	Must be in the NORM position.
ABNORM COND light	Indicates a power fault or overtemperature condition.
REMOTE light	In a multicabinet configuration, this light indicates that this cabinet is a slave and that power is controlled by the master cabinet. When unlit, it indicates that this is the master cabinet.

The DCP/25,30,35,50,55 models have three display status lights. The two rightmost lights (whose values are shown in Tables 2–9 and 2–14 during system booting and dumping) are either FF, for "idle", or a run number while the DCP/OS is operational. The leftmost light displays the operating modes (see Table 2–15).

Table 2–15. Operating Modes for the DCP/25,30,35,50,55

Value	Meaning
0	Reserved
1	Maintenance controller operational
2	Microcode loader operational
3	Offline diagnostics operational
4-9	Reserved
A	DCP/OS alternate user display
B	DCP/OS bootstrap
C	DCP/OS
D	DCP/OS Operational
E	DCP/OS
F	Hardware fault

Section 3
Console Mode Commands

This section does the following:

- Explains how to enter console mode
- Gives the console mode command syntax
- Describes each console mode command
- Explains how to send output to a printer

3.1. Using Console Mode

After a DCP is booted, all DCP/OS owned workstations are automatically set in console mode. Messages describing the progress of the initialization are displayed at these workstations. System initialization scans ports and puts disks and workstations in an up state. When installation is complete, a banner containing system information is broadcast to all workstations, commands are accepted, and programs may be started (see Figure 3–1).

```
DCP50        Id:DCP50-8   S/N: 508  Demand:  3/ 8 Bufpool: 8915- 78% Busy:  1%
11:34:00  On: lcp log              Batch:   2/ 7 Banks:    262- 94% Stat:NORM
 3=DEVNET: Enter command

▸T
▸  Active run: RCOOK /VWS-14
▸  Active run: RLJ   /VWS-15
▸  Active run: DEVNET/03        TELCON          208k
▸  Active run: VACUUM/04        PAUSE            16k
▸  Active run: RLJA  /VWS-16
▸RC RCOOK
▸  RCOOK ,X Run:01/Z  WS: VWS-14   Project: Q$Q$Q$ Start: 11:30:37 Status: OPEN
   Program:               Tasks: 0/ 0 SVCs: -----   Size:      0Kb  Ports:    0
▸RC RLJ
▸  RLJ      Run:02/Z  WS: VWS-15   Project: Q$Q$Q$ Start: 11:21:45 Status: OPEN
   Program:               Tasks: 0/ 0 SVCs: -----   Size:      0Kb  Ports:    0
▸RC DEVNET
▸  DEVNET   Run:03/Z  WS: -------- Project: DEVNET Start: 11:42:51 Status: ACTV
   Program: TELCON        Tasks: 2/20 SVCs: -----   Size:    208Kb  Ports:    1
▸SS

▸DCP50        Id:DCP50-8   S/N: 508  Demand:  3/ 8 Bufpool: 8906- 78% Busy:  1%
 11:33:59  On: lcp log              Batch:   2/ 7 Banks:    262- 94% Stat:NORM
```

017DG100.DRW

Figure 3–1. Example of DCP/OS Workstation in Console Mode

While in console mode, the system information banner (the top two lines in Figure 3–1) is refreshed periodically. The third line on this screen display is the solicitation for console input to the Telcon program under run DEVNET (Run 3); it is also refreshed periodically. All system activity messages are displayed and the screen scrolls up for each message.

These messages typically provide information pertaining to user jobs, or system recoverable error (SRE) messages. Refer to Appendix D for a complete list of messages.

Note: *If you have activated console mode by an @@CONS command or by signing on with @RUN,X, be sure to enter an @@END command before executing any program that solicits input.*

3.1.1. Command Syntax

When you specify parameters for a console mode command, you must separate the parameters from the command by at least one space. Multiple parameters are separated by commas.

3.1.2. Entering Commands

To enter a command, follow these steps:

1. Press the interrupt key (MSG WAIT). The screen scrolls up and displays a start-of-entry (SOE) character (▸).
2. Enter the desired command.
3. Press the transmit key (XMIT).

3.1.3. Sending Data to a Printer

You can obtain a hard copy of the data displayed at your workstation by using the NOPR and PRNT commands.

. Console Mode Commands

Table 3–1 lists the available console commands and briefly describes them.

Table 3–1. Console Mode Commands

Command	Command Description
ATT	Attach workstation to run
CA	Display current state of mass storage cache system
D	Display or set the current day, time, and date
DEB	Set workstation in debug mode with full-screen display
DN	Bring a facility down to a logical not-ready state
E	Error: terminate the program currently running for a specific run
FR	Free RAM disk space
FS	Display the status of a specific device or group of devices
II	Open dialogue with active run
LEV	Display the current system levels and the amount of memory installed
LOGO	Disable System Log File
LP	Display the load path or port path used for the most recent DCP boot

continued

Table 3–1. Console Mode Commands (cont.)

Command	Command Description
MV	Move a specific device to a specific port
MX	Set the maximum number of simultaneous jobs in batch or demand mode
NOPR	Stop the printing of data sent to the console
PRNT	Print data sent to the console
RC	Display the status of a specific run
RD	Display run status with a detailed analysis of current program memory usage
RES	Reset system parameters to the default settings
RT	Display run status with snapshot of currently active tasks
SNAP	Take dump of program while it is executing
SS	Display the status of the system
ST	Start a batch run
SX	Delete spool file
T	List all active runs
TB	Broadcast a message to all active workstations
TERM	Terminate a run
TM	Send a message to a specific workstation
UP	Bring a facility up to a logical ready state
X	Terminate the current active program of a specific run

1. ATT — Attach Workstation to Run

The ATT command attaches a workstation to a specified run. The specified run must be active and currently unattached (i.e., in batch mode or previously detached via @@DET). If the workstation is currently attached to a run, you must detach it with the @@DET command (see Section 5). If the run that you specify is not available for attachment, the workstation will remain detached from any run.

By using the ATT command, you can attach a workstation to a batch program and then proceed to debug the program using the @@DEB command (see Section 5).

Format

ATT *run-name*

Parameter

run-name is the name of any active run not attached to a workstation.

Example

```
ATT DEVNET
@@CONS RC
```

Response

```
▸ DEVNET   Run:03/Z  WS: VWS-12   Project: DEVNET Start: 16:24:54 Status: ACTV
▸ Program: TELCON        Tasks: 2/25 SVCs: -----   Size:    224Kb  Ports:    1
```

025DG100.DRW

Figure 3-2. ATT Command Output

3.2.2. CA — Cache Display and Control

The CA command without a parameter displays the current state of the mass storage cache system. The maximum number of banks used for caching may be specified.

The cache facility may be turned off by setting the maximum number of banks to zero. This does not automatically release banks currently in use for caching, but no additional banks will be assigned to the cache. Cache banks are automatically released for program use as needed.

Format

CA [*param*]

Parameters

IN	(Initialize)	purges all data from the cache banks and resets the cache statistics counters to zero.
RE		resets the cache statistics counters to zero.
	Note:	*For both IN and RE, the maximum and current number of banks used for caching are unchanged.*
n		specifies the maximum number of banks for cache (maximum is 62). The current number of cache banks is unchanged.

Example

```
CA
```

Response

```
▸          Mass-storage cache statistics since last reset (16:04:00)
   Max.    Current  Total    Seg.     File     Reads    Reads    Reads    Hit
   banks   banks    I/Os     loads    reads    > 1K     <= 1K    in cache rate
      53       11    3440       163     1410       34     1376     1370     99%
▸
```

026DG100.DRW

Figure 3–3. CA Command Output

3.2.3. D — Date

The D command displays or modifies the current date and time.

Format

D [*yy-mm-dd*] [*hh:mm:ss*]

Parameters

yy-mm-dd	modifies the curent date.
hh:mm:ss	modifies the current time.
none	displays the current date and time.

Examples

D
displays current date and time.

D 92-12-03 13:40:00
sets date and time. (It is not necessary to enter seconds when changing the time.)

D 92-12-03
sets date.

D 17:20:00
sets time.

4. DEB — Enter Debug

The DEB command sets your workstation to debug mode. See the *DCP/OS Programming Reference Manual* (7431 6894) for a description of the debug facility and commands.

Format

```
DEB
```

3.2.5. DN — Down Facility

The DN command changes the status of a facility to the not-ready (down) state. If you use the DN command for a disk device, all files cataloged on the volume are deregistered from the system catalog. You can recover the files by reentering the UP command for the device.

It is important to change the state of the device to DOWN and then to UP when you replace the disk or diskette. By using the DN command and UP command, you inform the DCP/OS that the disk is replaced.

Note: *It is not possible to down a disk drive that is in use. If you are unable to down a drive that does not appear to be in use, it may be because the drive contains the system log file (SYS$*SYSLOG) and logging is active. To deactivate system logging, refer to @TUNER in Section 4, or the console LOGO command in section 3.*

Format

```
DN device-name1[,device-name2,...]
```

Parameter

device-name is the name of a DCP/OS configured device (use FS,ALL if you are unsure what devices are configured).

Example

```
DN IF0
```

ɔ. E — Error: Terminate Program

The E command terminates the program currently executing under the specified run-name and generates an error, which produces an automatic dump if the necessary options are set (see the @CRASH command in section 4). If no program is active, no action is taken.

Format

E *run-name*

Parameter

run-name	is the name of any active run (there is no default).

Example

```
E MYRUN
```

3.2.7. FR — Free RAM Disk Space

The FR command frees the banks that are currently reserved for a RAM disk device and makes it available for other uses.

Format

FR *device-name1* [,*device-name2*,...]

Parameter

device-name is the name of the RAM disk device. (Use FS,MS to display the names of the mass storage devices if you are unsure of the name.)

Example

```
FR RAM1
```

8. FS — Facility Status

The FS command displays the status of a specific device or group of devices by generic type. Configured physical device names may vary from site to site. Generic type names are fixed.

Formats

1. FS *name1*[,*name2*,*name3*...]
2. FS,*type* [*n*]
3. FS,PORTS [*run-name*]

Parameters

types

ALL — displays all DCP/OS devices.

HOSTS — displays host channels.

IOM — displays all ports on the specified IOM.

MS — displays mass storage.

PORTS — displays all ports.

PP — displays the specified port.

WS — displays workstations.

names

Note: *The following are commonly used device names. It is also common practice to give volumes the same name as the device (not usually applicable to removable diskettes).*

DR*n* — indicates a specific removable cartridge disk.

DF*n* — indicates a specific fixed cartridge disk.

DT*n* — indicates a single density diskette.

DD*n*	indicates a dual density diskette.
WD*n*	indicates a BDBI (8409) Winchester.
SD*n*	indicates a SCSI (8441) diskette.
SW*n*	indicates a SCSI (8441) Winchester.
WS*n*	indicates a workstation.
CH*n*	indicates a channel.
RAM*n*	indicates a RAM disk.

Note: *The command FS,PORTS may optionally be followed by a run-name, which displays only the ports assigned to the specified run. FS,PP or FS,IOM is followed by a hexadecimal value indicating the port or IOM number, respectively.*

Example 1

FS,ALL
Displays all devices.

Response

```
▶@@CONS
▶FS,ALL
▶  IF0       Up    Port:0005 LM:09 MC:09 Ctl:7 Drv:2 Type:8441-D Vol:BOOT50
▶  LD$       Down  Port:001E
▶  WS$       Up    Port:0003 LM:67 MC:24 RID:23 SID:68
▶  SW0       Up    Port:0005 LM:09 MC:09 Ctl:7 Drv:0 Type:8441-W Vol:SW0    *
▶  WD0       Down  Port:0008 LM:07 MC:07 Ctl:0 Drv:0 Type:8409
▶  VWS-06    Up    VIRTUAL
▶  SW1       Up    Port:0015 LM:09 MC:09 Ctl:7 Drv:0 Type:8441-W Vol:SW2
▶  CH0       Up    Port:001E LM:11 MC:01 Host: INACTIVE
▶  SW2       Down  Port:0015 LM:09 MC:09 Ctl:7 Drv:0 Type:8441-W
▶  SW3       Up    Port:0015 LM:09 MC:09 Ctl:7 Drv:1 Type:8441-W Vol:SW3
▶  VWS-11    Up    VIRTUAL
▶  VWS-12    Up    VIRTUAL
▶  RAMDISK   Free  Size:    48Kb    Type:RAM
▶Ready
▶
```

027DG100.DRW

Figure 3–4. FS,ALL Command Output

Example 2

FS,MS
Displays all mass storage devices.

Response

```
▶@@CONS
▶FS,MS
▶  IFO       Up     Port:0005 LM:09 MC:09 Ctl:7 Drv:2 Type:8441-D Vol:BOOT50
▶  SW0       Up     Port:0005 LM:09 MC:09 Ctl:7 Drv:0 Type:8441-W Vol:SW0     *
▶  WD0       Down   Port:0008 LM:07 MC:07 Ctl:0 Drv:0 Type:8409
▶  SW1       Up     Port:0015 LM:09 MC:09 Ctl:7 Drv:0 Type:8441-W Vol:SW2
▶  SW2       Down   Port:0015 LM:09 MC:09 Ctl:7 Drv:0 Type:8441-W
▶  SW3       Up     Port:0015 LM:09 MC:09 Ctl:7 Drv:1 Type:8441-W Vol:SW3
▶  RAMDISK   Free   Size:   48Kb   Type:RAM
▶Ready
▶
```

028DG100.DRW

Figure 3-5. FS,MS Command Output

Example 3

FS SW0
Displays Winchester disk SW0.

Response

```
▶  SW0       Up     Port:0005 LM:09 MC:09 Ctl:7 Drv:0 Type:8441-W Vol:SW0     *
▶
```

029DG100.DRW

Figure 3-6. FS SW*n* Command Output

Example 4

FS WS0
Displays workstation WS0.

Response

```
▸FS WS0
▸  WS0       Up     Port:0003 LM:67 MC:24 RID:23 SID:68
▸
```

030DG100.DRW

Figure 3–7. FS WS*n* Command Output

Example 5

FS,PP 10
Displays port 10 information.

Response

```
▸   Port:0004 LM:6F MC:00 (---)  Asg'd: -
▸Ready
▸
```

031DG100.DRW

Figure 3–8. FS,PP *n* Command Output

Note: *For non-stop system (i.e., 2 partitions), the * designates that the same PP number in the other partition is assigned.*

3.2.9. II — Send Contingency to Program

The II command is used to send a contingency to the program executing under the specified run name, if the program has registered a contingency handler. For more information, see the section on contingency handling in the *DCP/OS Programming Reference Manual* (7431 6894). The II command is used to notify a program (especially one running in batch mode or detached mode) that you want to set up a dialogue. A program written to interact with workstations in the console mode will react to the II contingency for communicating with you using the available DCP/OS services. For more information on run services, see the *DCP/OS Programming Reference Manual* (7431 6894).

Format

II *run-name*

Parameter

run-name	is the name of the batch or detached run which gets the contingency. Use the T command to get a list of the active runs.

10. LEV — Level

The LEV command displays the system software, bootstrap, microcode, and release levels, and the amount of memory available in the DCP.

The available memory figure (displayed in kilobytes) should match the full complement. If it does not, this means the initial microcode load cannot access one or more memory arrays. This is not fatal to the running system, but does indicate a memory fault.

Format

LEV

Example

LEV

Response

```
▸ DCP/OS: 5R3.11        Boot: 5R3.4      Firmware:  15.013.000
  Memory:  4096Kb   Delivery: 5.3.11
▸
```

032DG100.DRW

Figure 3-9. LEV Command Output

3.2.11. LOGO — Disable System Log File

The LOGO command disables the System Log file and prevents the system from writing any further data to it. The parameter log disappears from the system status display when the file becomes disabled (this may or may not occur immediately).

This command is provided so that the disk on which the log file is currently active can be made available for maintenance (i.e., Format/Purge or any function which requires the disk to be down). To re-activate the system log file, use @TUNER (Section 4).

Format

```
LOGO
```

Example

```
LOGO
```

Response

```
Ready
```

12. LP — Load Path

The LP command displays the load path or port path that was used for the most recent DCP boot. See Section 2 for a description of load paths and port paths.

Format

LP

Example

LP

Response

```
▶@@cons
▶LP
▶  This system was booted with Load Path 01
▶
```

045DG100.DRW

Figure 3–10. LP Command Output

Note: *The response is different after the system has crashed, dumped, and rebooted.*

Example:

```
This system was booted with Load Path 01 initiated by Load Path 02
```

3.2.13. MV — Move Device to I/O Port

The MV command moves the specified device to the indicated port. The device must be in a down state before it can be moved.

Note that this command moves the device only for the life of the current boot and is not reflected in the **SYS$*SYSCFG** file, unless the online PDT/LPT is written to diskette using the @MONFIG utility.

Format

`MV device-name,port-number`

Parameters

device-name is any configured DCP/OS device.

port-number is any legal port number (hexadecimal).

Example

`MV SW0,4` moves the disk SW0 to port number 4.

l4. MX — Set Maximum Number of Jobs

The MX command sets the maximum number of jobs, either batch or demand, that can run simultaneously.

Format

MX D *nn* — sets maximum number of demand jobs.

MX B *nn* — sets maximum number of batch jobs.

Parameter

nn — is the number of jobs.

Note: *The total jobs active in the system may not exceed 15. However, the sum of the two maximum numbers that this command defines may exceed 15.*

Example

```
MX D 10
```

Response

```
▶DCP50        Id:DCP50-3   S/N: 503  Demand:  3/10 Bufpool:13123= 71% Busy:  2%
▶17:12:29  On: lcp log              Batch:   3/ 7 Banks:     37= 42% Stat:NORM
▶
```

033DG100.DRW

Figure 3-11. MX D *nn* Command Output

3.2.15. NOPR — Printer Off

The NOPR command cancels the printing of all console traffic on an auxiliary printer.

Format

```
NOPR
```

Example

```
NOPR
```

Response

```
READY
```

.16. PRNT — Printer On

The PRNT command prints a copy of all console traffic (excluding the periodic status display) on an auxiliary printer.

Format

PRNT [*number*]

Parameter

number	is the designated auxiliary printer. The default is 1 (equivalent to DID 73), and the range is 1 through 12.

3.2.17. RC — Run Check

The RC command displays the status of a specific run. The display includes the mode, the name of any active program, workstation/terminal, the number of service calls (SVCs) issued, the program size, and the number of ports assigned.

The SVC count often indicates the level of program activity.

The program size gives the number (in bytes) of 16K banks assigned to the program and does not include any buffers that are currently allocated.

Format

RC [*run-name*]

Parameter

run-name	is the name of any active run (defaults to your run-name).

Example

```
RC RCOOK
```

Response

```
▸ RCOOK    Run:02/Z  WS: VWS-12   Project: Q$Q$Q$ Start: 16:49:15 Status: OPEN
  Program:             Tasks: 0/ 0 SVCs: -----   Size:      0Kb  Ports:    0
▸
```

034DG100.DRW

Figure 3–12. RC Command Output

Note: *The SVC count will be displayed only if SVC counting has been enabled. To enable, see the @TUNER command in Section 4.*

3.2.18. RD — Query Run Status: Detailed

The RD command displays the same information as the RC command, and gives a detailed analysis of the system segment table.

If no program is active under the specified run, details are displayed for run zero (that is, the DCP/OS itself). When a program is active, only the user portion of the segment table is analyzed, thereby giving an accurate picture of the user system.

Format

RD [*run-name*]

Parameter

run-name	is the name of any active run (defaults to your run-name).

Example

RD DEVNET

Response

```
▸ DEVNET   Run:03/Z  WS: -------- Project: DEVNET Start: 16:24:54 Status: ACTV
  Program: TELCON        Tasks: 2/25 SVCs: -----   Size:    224Kb  Ports:    1
▸
   Segments:   Resident=   35 (   71Kb) Transient=  344 (  847Kb)

Res  (code):   Segs=   11 (   28Kb)
Res  (data):   Segs=   24 (   44Kb)

Tran (code):   Segs=  246 (  665Kb)  present=  198Kb in-use=   24Kb
Tran (data):   Segs=   98 (  182Kb)  present=   13Kb in-use=    6Kb
Tran (dynm):   Segs=   25 (   29Kb)

▸
```

035DG100.DRW

Figure 3–13. RD Command Output

Note: *The SVC count will be displayed only if SVC counting has been enabled. To enable SVC counting, see the @TUNER command in Section 4.*

3.2.19. RES — Reset Parameters

The RES command resets all adjustable system parameters, which might have been adjusted using the @TUNER utility program, to their original default values.

Format

```
RES
```

20. RT — Query Run Status: Tasks

The RT command displays the same information as the RC command and gives a snapshot of all the currently active tasks. Tasks can be suspended by calling DCP/OS System Service Calls (SVCs). Refer to the *DCP/OS Programming Reference Manual* (7431 6894) for more information on SVCs. The snapshot consists of the current procedure, address, stack depth, SVC type, and the first seven registers.

If a task is not suspended on an SVC, indicated by 0000 under the SVC heading, it was either time-sliced, is waiting on a lock, or is in debug.

The priority of each task (0 to 3) is displayed under Pri. A minus sign (-) following the priority number indicates the task was time-sliced.

A plus sign (+) following the priority indicates the task was involuntarily suspended due to PN4 conditions such as:

- buffer allocation
- segment loading
- locking and clear locking

Format

```
RT [run-name]
```

Parameter

run-name	is the name of any active run (defaults to your run-name).

Example

```
RT RCOOK
```

Response

```
▶@@CONS
▶RT DEVNET
▶  DEVNET   Run:03/Z  WS: -------- Project: DEVNET Start: 16:24:54 Status: ACTV
   Program: TELCON        Tasks: 2/25 SVCs: -----   Size:    224Kb  Ports:    1
▶  Task IPN   PN    Adr  PSW  #Stk SVC   R0   R1   R2   R3   R4   R5   R6   Pri
▶    4  00CC  00CC  FCF7 D35D   2  8304  0000 0000 0000 0000 0000 0000 3111  3
▶    7  00DB  00DB  0672 D35D   5  8212  0000 5000 3000 5000 5000 3000 0000  3
▶
```

036DG100.DRW

Figure 3–14. RT Command Ouput

Note: *The SVC count is displayed only if SVC counting has been enabled. To enable SVC counting, see the @TUNER command in Section 4.*

21. SNAP — Snapshot Dump

The SNAP command allows you to initiate a dump of a program while it is executing. The actual memory dumped depends on the type of dump that you have selected.

Format

```
SNAP run-name[,file-name[,type]]
```

Parameters

run-name is the name of any active run which is currently executing a program.

file-name is the name of the file (with optional qualifier) to which the dump will be written. The default can be set up by the run being dumped via @CRASH (see Section 4).

type is a character indicating whether a full memory dump or a program dump is requested:

P means take a program (plus DCP/OS) dump only.

F means take a full memory dump.

The default is P.

Example

```
SNAP  DEVNET,SYS$*SYSDUMP,P
```

Response

```
▶SNAP DEVNET,SYS$*SYSDUMP,P
▶DEVNET: Dumping...
▶DEVNET: Dump taken  to file SYS$  *SYSDUMP
```

040DG100.DRW

Figure 3–15. SNAP Command Output

Note: *If the system is in hard throttle, a machine dump will be taken instead of the intended program dump.*

22. SS — System Status

The SS command displays the status information that normally appears on the upper two lines of a console mode workstation. The information displayed includes the number of active runs and details of the memory buffer and bank pools, and the number of active control processors.

Format

SS

Example

SS

Response

```
▶@@CONS
▶SS
▶DCP50        Id:DCP50-3   S/N: 503  Demand:  3/10 Bufpool:15973= 86% Busy:  3%
 12:15:04  On: lcp log              Batch:   3/ 7 Banks:     71= 81% Stat:NORM
▶
```

037DG100.DRW

Figure 3–16. SS Command Output

Additional Explanation

System memory is divided into banks, buffers, and segments. Banks are 16K bytes; buffers are 128 bytes; segments are 4096 contiguous bytes. Banks and buffers are allocated and deallocated as needed; segments are only used by the system for instrumentation.

The available bufpool count is equal to the current buffers in the architectural buffer pools (both the zeroed and to-be-zeroed) plus the backup pool of buffers. The percentage available of buffers is equal to (the available bufpool count) divided by (the total buffers in the system minus the buffers in dynamic segments (if backup specified as % of total buffers)).

The available bank count is equal to the current banks available plus the current banks used by cache minus the minimum banks for cache. The percentage available of banks is equal to (the available bank count) divided by (the total banks in the system plus the current banks used by cache).

System status can be normal, soft throttle, or hard throttle. For more information, refer to Section 6.

The busy percentage is across all executing CPs. Demand and batch numbers represent the number of active runs out of the maximum runs allowed. The `On` field displays the number executing CPs (shown as 2cp in the above example), whether the `log` parameter is displayed (logging to the system log file (**SYS$*SYSLOG**) is turned on), and whether the `stats` parameter is displayed (statistics collection for system behaviour statistics is turned on (e.g., by @TUNER)).

23. ST — Start Batch Run

The ST command starts the run in the specified element. The first image in this element must be an @RUN statement.

Format

```
ST[,n] qual*file.element[,[RCW][,spool-size]]
```

Parameters

n	inhibits the creation of a spool file for this started batch job.
RCW	sets initial run control word (RCW) for the run to the specified hexadecimal value (maximum value is FFFF).
spool-size	specifies the size (in blocks) of the spool-file created for this started batch job (default is 128 blocks). The default spool file name is **SYS$*P$RUN0*n***: where *n* is the run number
*qual*file.element*	is the name of the symbolic element containing the runstream. If the file name is omitted, the DCP/OS searches the files **SYS$*SYSJOB**, **SYS$*SYSJOBX**, **SYS$*USRJOB**, and **SYS$*USRJOBX** for the element.

Example

```
ST MYQUAL*FILE.MYJOB
```

3.2.24. SX — Delete Spool File

The SX command closes the spool file currently open for a batch run and deletes that spool file.

If you are running a job in batch mode and the job fails, the data goes to a spool file. The name of this file is **SYS$*P$RUN*nn***. You can look at this file with an editor or with the @PRT command.

Format

SX *run-name*

Parameter

run-name	is the name of any active run (defaults to your run-name).

25. T — List Active Runs

The T command briefly displays all active runs, including the program name and size, if the program is active. The program size is shown to the right of the program name and gives (in bytes) the number of 16K byte banks assigned to the program.

Format

T

Example

T

Response

```
>@@CONS
>T
>  Active run: MARY  /VWS-06
>  Active run: RCOOK /VWS-12
>  Active run: DEVNET/03        TELCON          224k
>  Active run: VACUUM/04        PAUSE            16k
>  Active run: PAT   /VWS-11
>  Active run: BACKUP/09        PAUSE            16k
>
```

038DG100.DRW

Figure 3-17. T Command Output

3.2.26. TB — Terminal Broadcast

The TB command broadcasts a message to all workstations (both real and virtual) that are up and authorized for display. You can send up to a full screen.

Format

TB *message...*

Example

```
TB Celebrating cutover at 7 p.m.
```

27. TERM — Terminate Run

The TERM command terminates any active program under the specified run-name and then terminates the run. If no program is active, the run is terminated immediately.

Note: *The run, not just the program, is terminated. This is the main difference between the TERM command and the X command.*

Format

TERM *run-name*

Parameter

run-name	is the name of any active run (defaults to your run-name).

Example

```
TERM BOB
```

3.2.28. TM — Terminal Message

The TM command sends a message to a specific workstation (either real or virtual). Up to a full screen can be sent.

The message to the target workstation contains the name of the originating workstation.

The target workstation may be in console mode or demand mode.

Format

TM *workstation-name message...*

Example

```
TM VWS-21 Can you please free port 7?
```

.29. UP — Up Facility

The UP command changes the status of a facility to a ready state. If you use the UP command for a disk device, all cataloged files on that device are registered in the system catalog, unless a duplicate name is already registered. In that case, the file is unavailable. You cannot have two files with the same name on the system.

Multiple workstations may be configured on a single line and each workstation must be individually put into an up or down state.

The RID/SID address of each UTS workstation defaults to a general GID/DID at system boot and is picked up from the first transmitted message unless hard configured by @MONFIG.

Format

```
UP device-name1[,device-name2,...]
```

Parameters

device-name is any configured DCP/OS device.

Example

```
UP SW5
```
Brings up the SW5 device.

Response

```
▶@@CONS
▶UP SW3
▶  SW3      Up    Port:0015 LM:09 MC:09 Ctl:7 Drv:1 Type:8441-W Vol:SW3
  Warning:    1 Duplicate file(s) marked DOWN /   0 Temporary file(s)
▶
```

039DG100.DRW

Figure 3–18. UP SW*n* Command Output

30. X — Terminate Program

The X command terminates the program currently active under the specified run-name. If no program is active, no action is taken.

Note: *Only the program is terminated, not the run. This is the main difference between the X command and the TERM command.*

Format

X *run-name*

Parameter

run-name is the name of any active run (there is no default).

Example

DEVNET

Section 4
Demand Mode Commands

This section does the following:

- Tells you how to enter demand mode
- Gives the file name conventions and demand mode command syntax
- Describes each demand mode command
- Explains how to send output to a printer
- Tells how to execute programs
- Tells you what utility programs are available
- Explains how to use the utility programs
- Describes the function, format, and parameters for each utility program

4.1. Using Demand Mode

The following subsections describe how to use demand mode.

4.1.1. Entering Demand Mode

Demand mode provides an environment in which to execute programs (such as Telcon) and utilities. To enter demand mode from a DCP/OS workstation that is in console mode, or from a Telcon owned workstation that has a Telcon session to the target DCP/OS and is thereby a virtual workstation (VWS), you must initiate a demand run through the @RUN command. See the @RUN command in this section.

2. File Naming Conventions

File names have the following format:

```
qual*filename.element
```

Wherever *qual*file.element* is used in this section, the standard optional fields and defaults apply. See the @CAT command in this section for more information on qualifiers, file names, and elements.

3. Executing Programs

In demand mode, you execute programs by typing in the name of the *qualifier*file.element* containing the program you want to use. The @ symbol is used to invoke a demand mode program. The *qualifier*filename* defaults to **SYS$*SYSLIB**, the system file that contains all frequently used programs such as the DCP/OS utilities. If the program is not found in **SYS$*SYSLIB**, the system files **SYS$*SYSLIBX**, **SYS$*USRLIB**, and **SYS$*USRLIBX** are searched (if they exist).

Format

When a program is invoked in this manner, many options and parameters can be supplied on the command line, as shown in the following example:

```
@qualifier*file.element,ABCD....Z$!/xqt/xsst/nstk/useid
param1,param2,.........
```

*@qualifier*file.element*	is the name of the absolute *qualifier*file* and element containing the program you want to execute.
ABCD....Z	represents options A through Z. Any, all, or none of the options A through Z may be included on the command line.
$	forces a pseudotrap at the entry point to the program and enters debug mode.

!	forces the program to be completely memory-resident.
xqt	is the number of extra queue table (QT) entries reserved.
xsst	is the number of extra system segment table (SST) entries reserved.
nstk	is the number of task stacks available, up to a maximum of 32. The default number of tasks is defined by the program, or by the program builder (see the @BUILD RES command in this section).
useid	a three-character field that is used as the partitioned application use ownership of PPs and access to their LMs. Peer application in the other partition needs to have the same useid in order for both applications to have access through their respective PPs to the same LMs.
param-1,param-2	are the alphanumeric strings normally used to specify file and/or element names.

Notes:

1. *A program that runs out of stacks during execution may execute successfully when the number of stacks is increased. However, running out of stacks is usually an indication that the program is not running properly, or that the program designer did not allow for an adequate number of concurrent tasks.*
2. *You may enter the optional loader parameters* xqt, xsst, nstk *to extend or modify the size of the QT, SST, and the number of stacks (maximum concurrent user tasks). These parameters are normally set by the program or by the @BUILD or @DCPBUILD utilities and do not have to be specified when the program is executed.*

3. *If you have an error in your program, a dump may be required. Specify this action by executing the @CRASH utility program before you execute your program.*

4. Command Syntax

Demand mode and utility program commands are preceded by a control character (@). When parameters are specified on a demand mode command, they are separated from the command itself by at least one space. Multiple parameters are separated by commas.

5. Sending Output to a Printer

To send output to a printer in demand mode, use the demand mode bypass commands @@PRNT and @@NOPR, which are explained in Section 5.

4.2. Demand Mode Commands and Utility Programs

Demand mode commands are intrinsic commands. This means that they are actually part of the operating system. Utility programs, on the other hand, are not intrinsic; they are separate executable programs that are not part of the operating system.

Because demand mode commands and utility programs have the same format and are used in the same way, they are listed alphabetically in the subsections that follow. Each command function specifies whether it is a demand mode command or a utility program.

Table 4–1 summarizes demand mode commands.

Table 4–1. Demand Mode Commands

Command	Definition
@ADD	Add a command stream
@BRKPT	Breakpoint spool file
@END	End the input to the ELT program
@EOF	Indicate the end of data to a program
@FIN	Terminate a run
@PRIV	Change to privileged mode or change password
@PRT	Display file or element information
@RUN	Initiate a run (from console mode)
@START	Start a batch run

The DCP/OS utility programs are located in the file **SYS$*SYSLIB** and are executed from within a run. If you do not supply a specific file name, DCP/OS searches SYS$*SYSLIB for the specified utility program. If the utility is not found, the systems files **SYS$*SYSLIBX**, **SYS$*USRLIB**, and **SYS$*USRLIBX** are searched (if they exist).

Table 4–2 summarizes the standard utility programs in **SYS$*SYSLIB**.

Table 4–2. Standard Utility Programs in SYS$*SYSLIB

Utility	Definition
@ARCHIVE	Volume/File Archiver
@BUILD	Program builder
@CAT	Catalog a file (FUP)*
@CHG	Change a file or element (FUP)*
@COPY	Copy file or element (FUP)*
@CRASH	Dump specifier
@DELETE	Delete file or element (FUP)*
@DISK	Disk utility program (such as, format disk)
@DKCOPY	Diskette archive/restore
@DMPI	Telcon dump inspect
@DOWNLOAD	System downloader
@ED	Line editor
@ELSE	Conditional processor
@ELT	Element maintenance utility (FUP)*
@ENDIF	Conditional processor
@ERS	Erase program file (FUP)*
@FAC	Error code decoder
@FUP	File utility processor

continued

Table 4–2. Standard Utility Programs in SYS$*SYSLIB (cont.)

Utility	Definition
@HELP	Help document display
@IDUMP	Interactive dump inspect
@IF	Conditional processor
@IOL	Disk I/O error log display
@LIMODUMP	Line module dump
@LOG	JCL log message
@MODE	Set the disk type (FUP)
@MONFIG	DCP/OS configuration program
@MSG	Send message to operator
@PACK	Pack a program file (FUP)*
@PATCH	Absolute program patcher
@PORT	I/O port utility
@QUAL	Set assumed qualifier (FUP)*
@RCW	Display/modify run control word
@SPLIT	Split or combine files
@SYS	System display utility (time last booted, log file, behaviour statistics, etc.).
@TRCPP	Collect PP traces
@TUNER	Tune system parameters
@UBKEY	Display/modify user boot keys

* These commands are components of the File Utility Processor (FUP). You cannot copy these programs individually to disk. You must copy FUP.

1. @ADD — Add Command Stream

The @ADD demand mode command directs input of control statements and data read from the addstream element. If a program is active and has requested a read, the data read from the addstream is passed to the program in system message format. The program is not aware that the data originated in an addstream.

Addstreams may be nested to a maximum depth of eight by using @ADD commands within addstreams. However, addstreams may be chained together (and to themselves) by the C option. In this case, there is no nesting limit.

Note: *The C option closes the current add file before opening the chained add file. It does not return to the current add file upon completion of the chained add file.*

When the file name is not specified, the DCP/OS searches the files **SYS$*SYSJOB, SYS$*SYSJOBX, SYS$*USRJOB,** and **SYS$*USRJOBX** for the specified element.

Note: *The @FIN command is not executable from an addstream.*

Format

```
@ADD[,options] [qual*][file.]elt
```

Parameters

options

C	chains to new addstream.
D	allows the execution of @ADD as a control statement, in ELT,D mode.
E	implies @EOF after end of the addstream element.
L	lists control statements contained in the addstream while processing the addstream.

P prints this @ADD statement.

Example

```
@ADD,C MYQUAL*FILE.LOOPJOB
```

2. @ARCHIVE — Volume/File Archiver

The @ARCHIVE utility program lets you save a file, or even an entire volume in a compressed form. The file or volume can be restored later using the @ARCHIVE command. Compressed data can be stored as:

- An element in a file
- A file
- A sequence of one or more diskettes

@ARCHIVE can also verify that the compressed data matches the contents of the original file or volume being archived.

Format

```
@ARCHIVE[,options] [source,destination]
```

Parameters

required options (1 only):

S — Save the 'source' volume or file to the 'destination' device, file, or element.

R — Restore the 'destination' volume or file from the 'source' device, file, or element.

C — Compare the 'source' volume or file with the 'destination' device, file, or element.

other options:

D — Don't compress or expand — do a word-for-word copy/compare. This option is used where minimizing the time required for performing the archive procedure is more important than minimizing the amount of data space (number of data-diskettes).

F — Format archive data-diskettes as needed. This option is used only with the S option. It is particularly useful where the number of data-diskettes required is not known. If this option is not specified, any data-diskettes used must already have been formatted at 512 bytes/sector density. A purge is not required. This option is not supported on a DCP/5.

H — Display a help screen explaining how to use @ARCHIVE.

! — The 'memory-resident' option (!) can be used with any DCP program. It forces the program to be loaded entirely 'memory resident' and frees the load-file. It is particularly useful where @ARCHIVE is being executed from the hard-disk which it must 'down' in order to save.

source,destination

Normally, 'source' specifies where the data is to be read from, and 'destination', which must already exist, specifies where the data is to be written to. In the case of the 'C' option, however, 'source' specifies the original file or volume, and 'destination' specifies the diskette-device or the *file.element* where the archive-data is stored.

If the parameter contains a period (.), it specifies a file or element, in which case a qualifier (default or otherwise) may also be specified. Otherwise, it is the name of the diskette or hard-disk device. @ARCHIVE will 'down' any such device before using it (unless DCP/OS will not allow it to be downed).

Note: *In any case where a device-name is specified, only SCSI (8441) diskette or hard-disk devices are supported*

Examples:

Saving and Restoring a Volume

The following examples show how a volume can be saved, checked, and restored using @ARCHIVE:

Example 1

```
@ARCHIVE,SF! SWO,IFO
```

This statement is used to save the volume mounted on device SWO to one or more diskettes to be loaded in device IFO. Any data-diskettes required are formatted by @ARCHIVE before being written to.

Example 2

```
@ARCHIVE,C! SWO,IFO
```

This statement compares the contents of the volume mounted in device SWO with the archive-data read from one or more diskettes loaded in device IFO. It also verifies that the data written to the diskettes can be read and that it is a valid copy. This must be done before changing anything on SWO's volume.

Example 3

```
@ARCHIVE,R! IFO,SWO
```

This statement restores the volume mounted in device SWO from one or more diskettes loaded in device IFO. The media in device SWO must have been formatted at a time prior to this statement, but need not have been purged. @ARCHIVE does a purge internally. No pre-existing files are preserved. When this statement completes execution, the SWO device contains an exact copy of the archived volume, including any systems or dumps which may have been registered on the original archived volume (see @MONFIG).

Making Copies of a Boot-Diskette

The following examples demonstrate the use of @ARCHIVE in place of @DKCOPY to make a copy of a boot-diskette for a DCP/25. The D option is used to maximize speed. In this case, it takes about a minute to copy a boot-diskette.

Example 4

```
@ARCHIVE,SD IFO,CD5R2*BOOT.DCP25
```

This statement saves the boot-diskette volume mounted in device IFO to element DCP25 of file **CD5R3*BOOT**. This file is cataloged automatically with sufficient size before the statement is executed.

Example 5

```
@ARCHIVE,CD IFO,CD5R3*BOOT.DCP25
```

This statement compares the boot-diskette volume mounted in device IFO with the archive-data element (DCP25) in file **CD5R3*BOOT**. It also verifies that **CD5R3*BOOT.DCP25** is an exact copy of the entire boot-diskette.

Example 6

```
@ARCHIVE,RD CD5R2*BOOT.DCP25,IFO
```

This statement creates a boot-diskette volume on the diskette loaded in device IFO from the archive-data element DCP25 in file **CD5R2*BOOT**. The diskette loaded must be formatted (at 256 bytes/sector for a DCP/25), but need not have been purged.

Note: *The* F *option (which applies only to archive data-diskettes that would be of no use here) can only be used with its* S *option.*

Example 7

```
@ARCHIVE,CD IFO,CD5R3*BOOT.DCP25
```

This statement compares the boot-diskette created (in example 6) with the archive-data in element DCP25 of file **CD5R2*BOOT**. This verifies that the entire diskette can be read and that it is an exact copy of the formerly archived boot-diskette.

Backing-Up a File

The following examples show how a file can be saved in compressed format as an element in another file using @ARCHIVE. They also show the procedure for restoring a file from its compressed data element.

Example 8

```
@ARCHIVE,S MY*LIB.,SYS$*SAVE.12JUL92
```

This statement archives file **MY*LIB** to element 12JUL92 of file **SYS$*SAVE**. There must be sufficient room at the end of the **SYS$*SAVE** file for the compressed archive-data, or the statement will fail.

Example 9

```
@ARCHIVE,R SYS$*SAVE.12JUL92,MY*LIB.
```

This statement restores the file **MY*LIB** from the archive-data in element 12JUL92 of the SYS$*SAVE file. Since hard-disk I/O is more reliable than diskette I/O, there is probably no need to check the archive-data (see the C option), but you can if desired.

Saving and Restoring a System-Dump File

The following examples show how @ARCHIVE compresses a system dump file to one or more diskettes which can be sent to the development center for analysis. They also show how such a dump file can be restored (by the development center) from the compressed data diskettes.

Example 10

```
@ARCHIVE,SF SYS$*SYSDUMP.,IFO
```

This statement archives the **SYS$*SYSDUMP** file to one or more diskettes loaded in device IFO. Each diskette is formatted by @ARCHIVE before it is written to.

Example 11

```
@ARCHIVE,C SYS$*SYSDUMP.,IFO
```

This statement checks that the diskettes (written to in example 10) can be read (using device IFO) and that they are an exact copy of the **SYS$*SYSDUMP** file.

Example 12

```
@ARCHIVE,R SD1,SUPP*DUMP.
```

This statement restores the former **SYS$*SYSDUMP** file to file **SUPP*DUMP** from one or more archive-data diskettes loaded in device SD1. The file used (**SUPP*DUMP**) must be cataloged with a sufficient size before this statement is executed.

Note: *The filename does not need to be the same as the archived file.*

Comparing Files

When comparing a file on a hard disk to a file on one or more flexible diskettes using the D option, a file length mismatch may be reported. This occurs because the hard disk sector-size is 256 bytes, as opposed to the diskette sector-size of 512 bytes. This problem occurs only when the D option is used.

Example 13

```
@ARCHIVE,CD SYS$*STDFILE.,TESTFILE.
```

This statement compares the **SYS$*STDFILE** file with the **TESTFILE** file to see if their contents are identical. Because the D option is used, this is a word-for-word comparison (no compression/expansion is used).

Example 14

```
@ARCHIVE,CD DBASE.STDDATA,DBTEST.
```

This statement compares word-for-word (no compression or expansion is used) the contents of element **STDDATA** of the **DBASE** file with the contents of the **DBTEST** file.

Example 15

```
@ARCHIVE,C MY*LIB.,SYS$*SAVE.12JUL92.
```

This statement compares the compressed archive-data in element 12JUL92 of file **SYS$*SAVE** with the contents of the **MY*LIB** file.

Operational Considerations

In any case where a device name is specified, @ARCHIVE downs the device, then exculsively assigns the entire device for its own use. If it is unable to down the device, it is unable to assign it, causing error termination. This can happen if, for example, the system is booted from the hard disk, but the goal is to archive the hard disk volume. The hard disk device cannot be downed in this case because it is the 'system volume'. A 'load-path 0' boot from the boot-diskette avoids this problem. Also, system logging may be turned on, using the **SYS$*SYSLOG** file on the hard disk. If it is, system logging must be turned off (using either the LOGO console command or @TUNER) before the hard disk device can be downed. Finally, if @ARCHIVE is executed from **SYS$*SYSLIB**, and the memory-resident option (!) is not specified, the **SYS$*SYSLIB** file remains assigned and it is impossible to down the hard disk device, only if the file is on that device.

If the device containing the volume to be saved, restored, or checked was in the UP state before being used by @ARCHIVE, it is restored to the UP state when @ARCHIVE is through using it. However, a device used for data diskettes is left in the DOWN state.

Although @ARCHIVE is easy to use because it downs any specified devices, care must be taken to ensure that the right devices are specified, or valuable data on a device unintentionally specified could be destroyed.

In the case where a diskette device is used, @ARCHIVE prompts for diskettes to be mounted. Nothing is done with any diskette already in the drive until the prompt message is answered (although the drive's light may go on while the device is being exclusively assigned). The prompt message indicates which diskette (numbered 1 through N) is to be mounted. Processing can be aborted at any such prompt by pressing function-key 9 (F9). If you mount the wrong or a bad diskette, you can mount a different diskette, after which processing may proceed normally.

Any archive data diskette written to contains an ASCII-text label in sector 0, indicating the name of the file or volume that was archived, and the time and date the archive was initiated. For this reason, it is important to ensure that the time and date is set properly on the DCP prior to using @ARCHIVE. If for some reason the diskette is not labeled properly (using the sticker that goes on the diskette), the diskette 'label' can be displayed using the inspect volume function of @DISK.

When @ARCHIVE is finished, the `End Archive.` message is displayed.

3. @BRKPT — Breakpoint Spool File

The @BRKPT demand mode command closes the current spool file and assigns the new spool file (if the current and new spool file names are the same, any data in the current spool file will be lost). The spool file name and/or the spool file size (in blocks) may be specified. If omitted, the default values are used.

Format

```
@BRKPT    [file][,size]
```

Parameter

file — file name of the new spool file for this batch job (default is **SYS$*P$<run-name>**.

size — the spool file size in blocks (default is 128 blocks).

4.2.4. @BUILD — Program Builder

The @BUILD utility program uses module library file (MLF) elements, which are type omnibus elements, to create an absolute element in DCP/OS executable format.

MLFs are generated on the OS 1100 system and can be transferred to the DCP.

The technique that the @BUILD program uses to translate MLFs to DCP-loadable format is explained in the *DCP/OS Programming Reference Manual* (7431 6894).

Format

```
@BUILD[,options][qual*]file.[elt]
```

Parameters

options

B	suppresses full-screen mode.
D	forces program dictionary resident.
E	does not solicit input.
H	provides help information.
I	outputs absolute (ABS) to a file, not an element.
L	lists all modules as added (default is no list).
M	displays memory details.
N	displays minimal output.
R	sets read-only segments to transient and subsegmented.
S	builds a system utility processor (run must be privileged).

T provides a trace on each MLF (default is no trace).

Note: *The* L *option produces considerable detail on every MLF included in the build process, and slows down the execution of the @BUILD program.*

*qual*file.elt* is the name of the output element (ABS).

Note: *The element name may be omitted when you use the* I *option.*

Example

```
@BUILD,L      MON*LES.EXAM      specifies the o/p element.
LEV           Tuesday sys       is the user identifier.
IN            MON*LES.EXAM      is the input MLF element.
END                             ends the builder.
```

The @BUILD program is executed under the DCP/OS and not on the OS 1100 host. (See a description of the @DCPBUILD program in Appendix B.) The MLF element created as part of the collection process must first be copied from the OS 1100 host to the DCP.

Commands

The @BUILD program has several commands that you may use during the build process. These commands are listed in Table 4–3.

Table 4–3. @BUILD Internal Commands

Command	Explanation
ABORT	Terminates the @BUILD program
BASE	Searches DCP/OS dictionaries for Communications Processor Architecture (CPA) entries
END	Completes the build
EXCLUDE	Omits a specific module from the absolute
HELP	Displays help information
IN	Adds a specified MLF to the @BUILD
LEV	Stores level information
MEM	Forces memory type resident or transient
NOBASE	Inhibits search of DCP/OS dictionaries
RES	Specifies the size of CPA tables
UND	Displays currently undefined CPA entries

4.2.4.1. ABORT

The ABORT command terminates the @BUILD program immediately (no further output is written).

Format

ABORT

4.2. BASE

The BASE command allows the @BUILD program to search the DCP/OS dictionary for CPA entities. BASE is the default if the M option is not present on the call line. The BASE command is ignored if the M option is present on the call line. The BASE command is rarely needed for normal @BUILD functions.

Format

```
BASE
```

4.3. END

The END command tells the @BUILD program to complete the build process using the MLFs already supplied. The loadable output is written to the output file, and the @BUILD program terminates.

Format

```
END
```

4.4. EXCLUDE

The EXCLUDE command tells the @BUILD program to omit a specific module from the absolute (ABS), even if it is contained in one of the input MLFs.

EXCLUDE can also be used to define modules that are referenced but not in any MLF, and thus produce a tailored system with no undefined entries. The EXC statements must precede any reference to the excluded module.

Format

```
EXC,options module,[mod2][,mod3]...
```

Parameters

options

P is the procedure.

Q is the queue.

S is the segment.

module is the module name.

Example

```
EXC,Q QINP,QINP2,QINP3
```

4.2.4.5. HELP

The HELP command displays a text message that explains the commands in the @BUILD utility program.

Format

```
HELP command
```

Parameters

command is the command to be defined by HELP. The command can be any of the following:

```
[ABORT|BASE|END|EXCLUDE|HELP|IN|LEV|MEM|NOBASE|RES|UND]
```

4.6. IN

Each IN command causes the specified MLF to be added to the @BUILD program process.

If a module is already defined, you are asked whether to use the previously defined module (SKIP), replace it with this module (REPLACE), ignore the rest of this module (NEXT), or abort the build (EXIT).

Format

```
IN[,options] [qual*]file.[element],[qual*][file.[element]...]
```

Parameters

options

L	reads the MLF (to resolve undefined modules only).
R	forces segments resident.
T	forces segments transient.
qual	is the qualifier of the file containing the MLF.
file	is the name of the file containing the MLF.
elt	is the name of the MLF omnibus element (if omitted, MLF is assumed to be a file).

Note:

1. *An option is not necessary and is usually not used.*
2. *If an error occurs, you are asked if you want to continue. Respond with one of the following options:*

   C *means to continue.*

   X *means to exit (terminate the build immediately).*

Example

```
IN MYQUAL*FILE.MYELT
```

4.2.4.7. LEV

The LEV command stores the level information in the header record of the absolute system written to the output file. It may be used for identification purposes.

Format

```
LEV level information
```

Parameter

level information is up to 16 characters, including spaces.

Example

```
LEV TEST SYS APR 1
```

4.2.4.8. MEM

The MEM command forces the memory type for the named segments.

Format

```
MEM[,options] segment[,segment]
```

Parameter

options

R forces segment resident.

T forces segment transient.

segment is forced resident or transient, overriding the type specified on the SSTDEF.

4.9. NOBASE

The NOBASE command inhibits the @BUILD program from searching the DCP/OS dictionary for CPA entities. NOBASE is forced if the M option is present on the call line. The NOBASE command is rarely needed for normal @BUILD program functions.

Format

```
NOBASE
```

4.10. RES

The RES command specifies the size of CPA tables or the number of spare entries required in specific CPA tables. For example, if you know your program creates 100 dynamic segments at run time, specify 120 spare SST entries. This function is not required because DCP/OS dynamically creates the space (if available). However, the RES command ensures that sufficient space is available at load time.

Format

```
RES entity nnnn
```

Commands

The @BUILD program produces a header record containing values for CPA table sizes. If no RES directives are received, a default number of spare entries is created in each table. These sizes are used when the program is loaded. The default number of spare entries in each table is 10.
The default value for spare entries in each of these CPA tables can be changed by one of the RES commands listed in Table 4–4.

Table 4–4. RES Commands to Change Default Values

Command	Description
RES SST nnnn	Maximum number of entries is 8191
RES PT nnnn	Maximum number of entries is 2047.
RES PPPT nnnn	Maximum number of entries is 800.
RES QT nnnn	Maximum number of entries is 9999.
RES STK nnnn	Multiple processes may be active within a program, each requiring one stack. The default number of stacks generated is 1, but this RES command may alter the number (maximum 32). This value may be predefined by using the CPADEF MASM procedure. See the *DCP/OS Programming Reference Manual.*
RES BUF nnnn	The DCP/OS itself uses a 128-byte buffer pool. The default size of this pool is 512 buffers, but it can be changed by the RES command. If the BUILD is not a build of a DCP/OS system, this form of RES is not applicable.
RES LOCK nnnn	Allows you to specify the number of lock table entries. The default is 128. The maximum number of entries is 8192.

Note: *In all forms, the numeric field on the RES command is a maximum of 4 decimal characters, or a smaller limit according to the particular CPA table maximum.*

Table 4–5 lists the various maximums for structures at build and execution time.

Table 4–5. Structure Maximums

	@BUILD	@DCPBUILD	SVC (run time)
Max QL Entries	1024	1024	8192
Max LA Entries	16382	16382	32766
Max QT Entries			10240

Examples

```
RES STK 10
```
specifies up to 10 concurrent tasks.

```
RES SST 120
```
specifies 120 spare segment table entries.

4.11. UND

The UND command causes a list of all currently undefined CPA entities to be displayed. It can be entered whenever a command prompt is displayed.

Format

```
UND
```

4.2.5. @CAT — Catalog File

The @CAT utility program, which resides within the FUP, catalogs a file on mass storage. The mass storage volume can be specified either by generic device type (8406, 8408, and so forth) or by specific volume name. Specifying a volume overrides any mass storage generic type. If neither disk type nor volume name is specified, the system attempts to catalog a file on the system volume first, then on any available disk.

Format

```
@CAT[,options] [qual*]file.,[[type][///size],[volume]]
```

Parameters

options

D	takes file offline (down). A different file with the same *qual*filename* can then be cataloged. A down file is treated as if it were nonexistent.
F	catalogs the file as full (highest block written = *size*).
H	provides help information.
S	switches the file (brings down this file and brings up another file that has the same *qual*file*); the volume name is required.
T	catalogs the file as temporary (all temporary files are deleted when the run is terminated).
U	brings up the file (marks a file that was down as currently available). Only one file with a given *qual*file* can be up at any time. All file operations using this *qual*file* assume the up file; the volume name is required.

qual is the optional qualifier of the file to be cataloged. Maximum of six characters including alphabetic (A through Z), numeric (0 through 9), dollar ($), or hyphen (-). If no qualifier is given, the default value is the current project-ID. If a asterisk (*) is entered without an explicit qualifier, the default is the assumed qualifier. See the @QUAL command.

file is the name of file to be cataloged. Maximum of eight characters including alphabetic (A through Z), numeric (0 through 9), dollar ($), or hyphen (-).

type is the mass storage type, as follows:

D	is any fixed disk (default)
DD	is 8406 FDDS diskette
DF	is 8408 cartridge, fixed
DR	is 8408 cartridge, removable
DT	is 8406 FDS diskette
IF	is IFD diskette
RAM	is RAM disk
SD	is 8441 diskette
SW	is 8441 Winchester
WD	is 8409 Winchester

size is the number of blocks in file (default is 128).

volume is the name of volume on which to allocate the new file for F, T, and no options), defaulting to the first available space on a drive other than a removable diskette. The volume name is required for the S and U options.

Example

`@CAT,T MYFILE.` creates a temporary file (deleted when run ends).

Note: *An @ character is allowed in the* qual *and* file *names, but it is recommended to avoid its use because other utilities operating on them may fail and/or give strange status messages.*

4.2.6. @CHG — Change File or Element

The @CHG utility program resides within the FUP. It changes the read-only attributes of files or the names of qualifiers, files, or elements.

Format

```
@CHG,[options] file1.[elt1],[file2.[elt2]]
```

Parameters

options

A,O,R,S	Element type options corresponding to absolute, omnibus, relocatable, or symbolic, respectively. Any or all of these options may be present for an element name change. If more than one element type option is present, an attempt is made to rename the specified element of each of the indicated types. Each successful name change is displayed for your information. At least one element type option must be present for an element name change.
F	indicates that the file name is to be changed and requires that the user is privileged.
H	is help information.
V	indicates that the file is to be marked read-only and requires that the user be privileged.
Z	requests that the read-only attribute of the file be cleared to allow write access. This requires that the user be privileged.
file1	is the original *qual*file* to be renamed or altered.
elt1	is the original name of the element to be renamed. At least one of the A, O, R, S options is required.

file2 is the new *qual*file* when you specify the F option (file name change).

If you do not use the F option when specifying an element name change, this parameter is optional and defaults to *file1*. If *file2* is specified, it must be the same as *file1*.

elt2 is the new name for the element being renamed (A, O, R, S options).

Note: *The file and element names cannot both be changed with a single @CHG command. However, the V and Z options can both be used with either a file name change or an element name change, provided that the user is privileged.*

Examples

`@CHG,F Q$Q$Q$*FILEA.,SYS$*FILEB.`	changes qualifier and name.
`@CHG,ZFV MON*SYSLIB.,SYS$*SYSLIB.`	clears read-only, changes qualifier, and sets read-only.
`@CHG,AR MON*SFILE.E1,.E2`	renames absolute and relocatable elements from E1 to E2.
`@CHG,AV LIBFILE.TEST,.PRGM`	renames absolute element TEST to PRGM and changes LIBFILE (under default qualifier) to read-only.

4.2.7. @COPY — Copy File or Element

The @COPY utility program resides within the FUP. It copies a file or element from a source location to a destination location. The source and destination is the DCP or a channel attached OS 1100 system. If no options are specified, the complete source file is copied to the destination file.

Format

```
@COPY[,options] [chan1:][qual1*]file1.[elt1],[chan2:][qual2*]
file2.[elt2]
```

Parameters

options

A,O,R,S — are element type options, corresponding to absolute, omnibus, relocatable, or symbolic, respectively. They are all allowed in any combination unless the I option is used, in which case only one element type option is allowed. The absence of an element type option implies file-to-file copy and precludes use of the I option.

H — is help information.

I — indicates that a file is to be created from an element, or an element is to be created from a file. This also inhibits the normal default action of forcing an omitted *elt2* to be equal to *elt1*. It is mutually exclusive with the P option.

L — lists each element name during a generic copy.

P — is the generic copy indicator. All elements satisfying the input specifications are copied to *file2*. The input specification is made up of element type option, *chan1*, *qual1*, *file1*, and *elt1*. It is mutually exclusive with the I option.

Destination element types and names are the same as source types and names.

chan1 is the name of the source host PDT channel name (optional). You cannot copy host-to-host from a DCP.

qual1 is the qualifier of the source file (optional). If no qualifier is given, the default value is the current project ID. If an asterisk (*) is entered, the default is the assumed qualifier. See the @QUAL command in this section. If the file is located on the DCP, the qualifier may be 1 to 6 characters long. If the file is located on a host instead of the DCP, the qualifier may be 1 to 12 characters long.

file1 is the name of the source file (required).

elt1 is the name of the source element (optional).

chan2 is the name of the destination host PDT channel name (optional). You cannot copy host to host from a DCP. The OS 1100 host channel name is the PDT name of a channel connected to the host. DCPFT must be running on the OS 1100 host and have the channel assigned. The channel must be up on the DCP. When transfers are not being performed, the channel should be down at the DCP (for BMux channel).

qual2 is the name of the destination file's qualifier (optional). If no qualifier is given, the default value is the current project-ID. If an asterisk (*) is entered, the default is the assumed qualifier. See the @QUAL command in this section. If the file is located on the DCP, the qualifier may be 1 to 6 characters long. If the file is located on a host instead of the DCP, the qualifier may be 1 to 12 characters long.

file2 is the name of the destination file (required).

elt2 is the name of the destination element (optional). The name in *elt1* is used for *elt2* when a destination element name is omitted but is needed (element type specified, without I or P option).

Examples

`@COPY MYCONFIG.,CONFIG.`	copies file-to-file using the default qualifier (project-ID).
`@COPY,A SYSLIB.DISK,RENE.`	copies absolute element-to-element using the default *elt2* to be equal to *elt1* and the default qualifiers *qual1* and *qual2* equal to the project-ID.
`@COPY,O SYSLIB.DISK,KK.OLDDISK`	copies omnibus element-to-element renaming the element using the project-ID as the default *qual2*.
`@COPY,IA TELCON.TELCON,*TEL.`	copies element-to-file using the assumed qualifier as the default *qual2*.

). @CRASH — Error Action Registration Program

The @CRASH utility program allows you to define a dump file by name and to request specific dumping action when there is an error in the program. The file is not normally created when @CRASH is invoked; it only sets parameters so that if a program executed later aborts, a suitably sized dump file can later be cataloged, if needed, and then if the file exists, a dump is initiated to the selected file. By option C, @CRASH will attempt to catalog the needed dump file now to allow the runstream to take corrective action if a dump file cannot be created. It also limits the number of multiple dump files in use in low mass storage situations. You can analyze a dump file by using the @DMPI or @IDUMP utilities, or by copying the dump file to an OS 1100 host, thereby allowing you to use the @DCPDUMP program.

Format

```
@CRASH[,options] [qual*][file.][,{load-path,number}]
```

Parameters

options

B	boots the DCP on detecting an error. The load path is required. If the load path is zero, the current top cap settings are used; otherwise, the hexadecimal value defines the load path to use.
C	catalogs a dump file with sufficient size to hold the largest possible dump.
D	enters debug on a trap.
F	performs a full dump (all memory).
G	generates a unique dump file name.
H	provides help information.
N	disables dumping and debug on trap.

P dumps user program and DCP/OS memory.

U uses multiple dump files, where *number* is the highest numbered filename of the form **$$$$$$$***N*, where *n* is from 0 to number (default is 9).

qual is the qualifier of the dump file (optional).

file is the name of the file to be used for the dump. If it exists at the time of the error, it is used. It must be large enough to accept the dump. If it does not exist, it is cataloged by DCP/OS. The file is not required if you use the D, G, H, or N options. The default *qual*file* is *projectid****D$R***n***$$$$**, where *n* is the job number.

A warning is displayed if the file contains a dump that has not cleared (see @DMPI,C). If the file does not exist, it is cataloged. If the file exists but is not large enough, it is deleted and recataloged.

Example 1:

```
@CRASH,UP file2
```

Example 2:

```
@CRASH,CP DUMPF.
@IF,FN    DUMPF.
@MSG,C    UNABLE TO CREATE DUMP FILE
@ENDIF
```

Additional Discussion

The error action registration is effective until an error dump occurs or CRASH,N is performed. Taking SNAP dumps or DEBUG D command dumps does not clear the registration. However, they may change the extent of the dump registration (F or P) and the file to which it is taken.

For debugging purposes, PD is a useful option combination. It forces debug to be entered upon encountering a trap (D option) and also allows you to perform a dump (P option) while in debug. See the *DCP/OS Programming Reference Manual* (7431 6894) for more information on the debug dump command.

The B option is primarily intended for unattended DCPs with no workstations attached.

Caution

The B option forces a reboot of the DCP when a program error is encountered and must therefore be used with caution, especially in a multiuser environment.

Note: *In some cases, such as in system hard throttling, there may not be enough buffers for the dump procedure, resulting in a lost dump.*

4.2.9. @DELETE — Delete File or Element

The @DELETE utility program resides within the FUP. It deletes a file or element.

Format

```
@DELETE[,options] [qual*]file.[elt][,[qual2*]file2.[elt2]]
```

Parameters

options

A,O,R,S	are the element type options, corresponding to absolute, omnibus, relocatable, or symbolic, respectively. The absence of an element type option implies that a file is to be deleted.
H	is help information.
qual	is the qualifier of the file to be deleted (optional).
file	is the name of the file to be deleted, or the name of the file containing the element to be deleted (required).
elt	is the name of the element to be deleted. It is required if the element type is specified; otherwise, it is not permitted.

Examples

`@DELETE,S SYSJOB.MYJOB`	deletes a symbolic element with the default qualifier equal to the current project ID.
`@DELETE MYQUAL*FILE.`	deletes a file.

10. @DISK — Disk Utility Program

The @DISK utility program allows you to select disk utility functions from a menu. When you have entered the @DISK program, you can select one of the functions listed in Table 4–6.

Table 4–6. @DISK Functions

Function	Description
Format disk	Initializes control information and data on a disk
Inspect volume	Inspects the contents of a disk
Patch volume	Changes the contents of a disk
DSL display	Displays the data set labels (DSLs) of a disk
Rename volume	Changes the name of a disk
Verify volume space table	Checks the volume space table (VST) and displays volume map
Backup volume	Copies the contents of one disk onto another disk
Rename file	Changes the name of a file on a disk
Purge disk	Resets the control information only (delete all files) on a disk
Pack volume	Repositions files on volume to free up space
Badspot disk	Adds badspot area to disk error map

Format

```
@DISK[,options]
```

Parameters

options

B is batch mode (scrolled terminal display).

H is help information.

Format Disk

To format a disk, use the procedure in Table 4–7. The device must be down while a format is being performed.

Table 4–7. Format Disk Procedure

Step	Action	Results
1	Choose interlace or sector density	The disk is formatted.
2	Enter volume label name.	Control tracks are written to the disk along with the appropriate number of empty DSLs. Calculations and allowances for the different sizes and types of disk media are done automatically.

Inspect Volume

The inspect volume function is used to inspect the contents of a volume on a particular disk drive. The function initially displays the contents of sector 0, and then can be asked to display any valid sector. The sector number specification can be in decimal or in hexadecimal notation. You can display the next or the previous sector. The display indicates the device name and sector number in both decimal and hexadecimal notation.

The data is displayed in a common inspect and change mode format, which shows the data in hexadecimal and, where possible, in ASCII character form. The inspect volume can be used on a device that is up or down because it is a read-only function.

Both the inspect and patch functions display data in sectors. Table 4–8 shows the normal sector densities for supported mass storage.

Table 4–8. Normal Sector Densities for Mass Storage

Model	Disk Type	Bytes/Sector
8406-0	FDS	128
8406-4	FDDS	128 or 256
8408	Cartridge	256
8409	Winchester	256
8441	Winchester	256
8441	5 1/4-inch	256
8441	5 1/4-inch IFD	512
8441	3 1/2-inch	256
8441	3 1/2-inch	512
8-inch	8-inch IFD diskette	128

Patch Volume

The patch volume function has the same features as the inspect volume function but it also lets you alter the contents of any sector by changing the screen data and transmitting the screen back to disk. The change must be made to the hexadecimal (not the ASCII character) representation of the data. Because patch volume is a write function, the device must be down when the patch volume function is selected.

The hexadecimal display shows lines with eight entries of four hexadecimal characters:

```
1234    5678 ABCD    FFFF    1122 6677 DEAD    0099
```

To make a change, change the hexadecimal value or enter an ASCII character preceded by a space, as follows:

```
ABCD   1234  T Y P E        1234 1234 1234
```

DSL Display

The DSL display function displays the active data set labels (DSLs) on a disk. After you specify the device, a full screen of DSLs is displayed that shows file name, file base block number, file size, and highest block written. Additional pages of active DSLs can be displayed on demand, until all are viewed. The DSL display function can be used on a device that is up or down because it is a read-only function.

Rename Volume

The rename volume function allows you to rename a volume. After you enter the device name, the current volume name is displayed and you are prompted for a new volume name. The device must be down during execution of the rename volume function.

Verify Volume Space Table

The verify volume space table function displays the in-use and free space on a specific volume. This display may be used to match the display of DSLs to verify space allocation.

Overlay records and orphan records are also displayed. Overlay records are those included on more than one DSL or space item. Orphan records are those not defined by a DSL or by a space item. The device may be either up or down during execution of this function.

Backup Volume

The backup volume function allows you to copy the contents of a volume to another volume. To copy a volume, follow these steps:

1. Enter the name of the source device. The device may be up or down. If the source and destination devices are not compatible, the destination device is rejected.
2. Enter the name of the destination device. The device must be down. If the source and destination devices are compatible, you are given a regular progress report while the copy is being performed.

Note: The copy routine is made in track-size transfers or in smaller transfers, where the size of a track exceeds the maximum I/O size limit.

Rename File

The rename file function allows you to rename the file name field of a data set label (DSL). To rename a file, follow these steps:

1. Enter the device and current file name. The DSLs are read until the requested file name is found. The DSL is then displayed.
2. Enter the new name for the file. The DSL is rewritten to the disk.

Notes:

1. *If the device is down during execution, the new file name becomes active the next time the device is brought up.*
2. *If the device is up during execution, the @DISK program needs to assign the device for update. Once updated, the new file name becomes effective immediately.*

Purge Disk

The purge disk function allows you to purge a disk. It is similar to the format disk function. It is faster because this function is used on a disk previously formatted by @DISK. All you must do is rewrite the control information and DSLs.

Caution

All data on the disk is lost when the unused DSLs are written to the disk.

Pack Volume

The pack volume function moves the data extents of active data set labels (DSLs) to the lower portion of the disk, and updates the DLSs to reflect the new data extents. The net result is that files are packed to the lower portion of the disk, the disk is no longer fragmented, and unused space is contiguous.

Bad Spot Disk

The bad spot disk function allows you to note bad areas of the disk in the disk error map. If an I/O error occurs, for example, use the following steps:

1. Use @IOL to determine whether a bad area of disk is indicated. Note the sector number.
2. Use `verify-VST` to identify the name of the bad file.
3. Copy the file to another file.
4. Delete the bad file.
5. Take the disk down.

Note: *If this disk is the system volume, the DCP must be reloaded so that this is not the system volume. Refer to @MONFIG for more information.*

6. Use `Bad spot disk` to mark bad spots in the sectors. You can use `Pack volume` at this point to free disk space, though it is not essential.
7. Bring the disk up.

1. @DKCOPY — Diskette Archive/Restore

The @DKCOPY utility program copies a complete diskette to a file on mass storage or copies an archive file back to a diskette. Diskette to diskette copying is not supported.

Format

@DKCOPY[,*options*] *in*,*out*[,[*fblk*]/[*lblk*]]

Parameters

options

H	displays help information.
N	displays minimal output.
V	verifies the copy.

in is the diskette device identifier (no period) or a DCP filename.

out is the DCP filename or the diskette device identifier.

fblk is the first block to be copied (default is 0).

lblk is the last block to be copied (the default is the last block on the diskette).

Note: *The diskette device ID is entered as an element name without a trailing period.*

Examples

@DKCOPY IF0,MON*IF0COPY.	archives all the contents of a diskette in drive IF0 to the file **MON*IF0COPY**.

`@DKCOPY MON*IF0COPY.,IF0`	copies the archive file **MON*IF0COPY** back to a suitably formatted diskette in drive **IF0**.

Note: *You do not need to catalog the output file before you use it. The* `@DKCOPY` *command catalogs the file for you.*

12. @DMPI — Dump Inspect

The @DMPI utility program inspects a dump initiated to DCP mass storage.

Format

```
@DMPI[,options] [qual*]file.elt
```

Parameters

options

A — analyzes dump file *qual*file.elt*.

B — provides debug mode output. (This option can be used only in conjunction with the A option.)

C — frees dump file *qual*file.elt* for reuse. This does not imply that a second dump cannot go into the file unless cleared. A second dump goes into the file without the need to clear the 1st dump.

D — deletes free dump files.

H — displays help information.

N — omits banner lines.

R — reports file status (default if A, C, and U are omitted).

U — sets in-use flag for file *qual*file.elt*.

qual — is the qualifier of *file* (optional). Qualifier conventions apply.

file — is the file name (or root file name for the D or R options) to be used.

elt — is the element name.

Example

```
@DMPI.A  MYQUAL*DUMP.
```

13. @DOWNLOAD — System Download

The @DOWNLOAD utility program is used as an intrinsic component of download booting. See the *Communications Delivery Installation Guide* (7831 5645). This utility can also be used in interactive mode where you can control many actions that occur automatically at boot time. For example, if the download file already exists, you are asked whether or not it should be overwritten.

Format

```
@DOWNLOAD[,options] [,device[/dir]]...[,device[/dir]]
```

Parameters

options

C	provides controlled load (presents dialogue to bring up or format the disk).
D	processes disk only (does not load, but recovers specified device).
F	catalogs the system dump file **SYS$*SYSDUMP**.
H	displays help information.
L	displays long listing.
N	inhibits configuration update.
T	sets trace mode.

device is the physical device table (PDT) name of configured mass storage, the first device specified is the system destination.

dir is the mass storage device directive, where:

F Format the mass storage device.

P Purge the mass storage device.

The format/purge boot keys, specified on a DOWNLOAD load path, take precedence over the @DOWNLOAD device directives. The @DOWNLOAD device directives will only apply when the format/purge boot keys are marked as off.

Example 1

```
@DOWNLOAD,LT
```

The T (trace) option can be used in this mode for troubleshooting. It allows tracing of all data blocks received.

Example 2

```
@DOWNLOAD SW0/F, SW1/P, SW3/P
```

This system will be downloaded to device SW0, after device SW0 has been formatted and devices SW1 & SW3 have been purged.

*Note: The @DOWNLOAD program resides in the **SYS$*SYSLIBX** file on the delivered boot diskette. Unless you modify the installation routine, @DOWNLOAD is not available in the normal system library. To run @DOWNLOAD, be sure that the boot diskette is in a suitable drive and that the drive is UP. Exercise care in running @DOWNLOAD interactively since you can purge useful mass storage.*

14. @ED — Line Editor

The @ED utility program is a symbolic text editor. For more information, refer to the *OS 1100 Text Editor Operations Reference Manual*. The default size can process up to 1,575 ASCII records.

Format

```
@ED[,options] [name1][,name2]
```

Parameters

The following options are deviations from the *Text Editor Operations Reference Manual*:

options

G	is the giant-size editor (7,975 records).
H	displays help information.
M	is the medium-size editor (4,775 records).
S	is the super-size editor (11,175 records).
T	is used to translate TPF$ to T@RUN-ID.

Example

```
@ED,G MYQUAL*FILE.MYELT
```

4.2.15. @ELSE — Else Condition Processor

The @ELSE utility program is a conditional processor for addstreams. The statements between the @ELSE and the paired @ENDIF are processed or conditionally skipped as a result of the preceding @IF processor call. Refer to the @IF command for more information on the @ELSE command.

Format

```
@ELSE
```

16. @ELT — Element Maintenance Utility

The @ELT utility program resides within the FUP. It generates or prints a symbolic element in a file. The @ELT program can only count the number of lines in an element (no options are available).

Format

```
@ELT[,options] [qual*]file.elt[,sentinel]
```

Parameters

options

D	allows inclusion of job control language (JCL) commands in the input. The JCL commands normally terminate ELT, but with the D option, only an @END terminates ELT. If a sentinel is specified, only an @END with the specified sentinel starting at character 5 terminates ELT. The D option is only applicable if the I option is included.
H	displays help information.
I	creates the specified symbolic element in the specified file. Absence of the I option indicates that a list of an existing symbolic element is required.
L	lists a symbolic element.

qual	is the qualifier of *file*. Default conventions apply. See the @QUAL command.
file	is the name of the file for insertion of the element or name of the file containing element to print.
elt	is the name of the symbolic element to insert or to print.
sentinel	is a 1- to 4-character sentinel for use with D option.

Example

```
@ELT,ID SYS$*SYSJOB.DAN
@RUN MYJOB
@DKCOPY IF0,IF0COPY.
@FIN
@END
```

17. @END — End Of @ELT Input

The @END demand mode command terminates input to the ELT utility program when used in data mode (for example, @ELT,ID).

See the @ELT command described in section 4.

Format

@END,[*sentinel*]

Parameters

sentinel is a 1- to 4-character sentinel used in the @ELT statement.

4.2.18. @ENDIF — End If Condition

The @ENDIF utility program is a conditional processor for addstreams. It delimits the scope of an @IF - @ELSE structure. If runstream statements are skipped because of a prior @IF or @ELSE statement, the skipping is terminated when the @ENDIF statement is encountered.

Refer to the @IF command for more information.

Format

```
@ENDIF
```

19. @EOF — End of File

The @EOF demand mode command indicates the end of data to a program.

Format

```
@EOF
```

4.2.20. @ERS — Erase Program File

The @ERS utility program erases a file and formats it as a program file. This creates an empty table of contents (TOC) in the specified file, preparing it for the insertion of elements.

Format

```
@ERS[,options] [qual*]file.[[,qual2*]file2.[,[qual3*]file3.]]
```

Parameters

options

H	displays help information.
qual	is the qualifier of file. Default conventions apply. See the @QUAL command.
file	is the name of the file to be erased.

Example

```
@ERS MYQUAL*FILE.
```

21. @FAC — Error Code Interpretation Program

The @FAC utility program interprets an error code returned from DCP/OS. See Appendix A for DCP/OS error codes.

Format

```
@FAC[,options] err-code
```

Parameters

options

H	displays help information.
P	interprets CPA error as a port processor (PP) type error.

err-code signals to enter one of the following error code type numbers: *nnnn* or *n-nn*.

nnnn	is a system error shown as a 4-digit hexadecimal number.
n-nn	is a CPA error code, type or subtype (with the numbers expressed as hexadecimal numbers).

Examples

```
@FAC      9013
@FAC      3-05
@FAC,P  4-16
```

4.2.22. @FIN — Terminate Run

The @FIN demand mode command terminates an active run. The workstation then reverts to console mode unless it is a virtual workstation, in which case the TELCON session is closed.

In a batch mode runstream, the @FIN command terminates the run.

Format

```
@FIN
```

3. @FUP — File Utility Processor

'he @FUP program is invoked by the operating system to execute all of he following commands:

@CAT
@CHG
@COPY
@DELETE
@ELT
@ERS
@MODE
@PACK
@QUAL

'hough the @FUP command displays a summary of these commands, you annot execute any of the commands without having the FUP program ıvailable.

Format

)FUP[,H]

Parameters

)ptions

H displays help information.

4.2.24. @HELP — Help Program

The @HELP utility program is used to obtain information about using DCP/OS and executing a batch or demand run. This menu driven program allows you to obtain information about particular topics. It gives you answers to questions dynamically (online) that you would otherwise have to look up in a manual such as this.

Format

```
@HELP[,options] [[qual*]file.][elt]
```

Parameters

options

H — isplays information about running the HELP processor and then exits.

0 — uses HELP-ORM (obtained from COMUS INFO file) for the HELP information element name.

P — uses HELP-PRM (obtained from COMUS INFO file) for the HELP information element name.

qual — is the qualifier of *file*. Default conventions apply. See the @QUAL command.

file — is the file from which the HELP information is to be obtained. If not specified, **SYS$*SYSLIB** is assumed.

elt — is the element name of the HELP information element. If not specified, HELP is used as this element (except in the case where the 0 or P option is used as described above).

Note: *Users may generate their own HELP information elements using the @DOC processor under the OS 1100 system. After copying the @DOC processor output file to an element on the DCP, you may access it with the HELP processor.*

Example

```
@HELP
```

Additional Discussion

This processor call accesses the system general HELP information from the **SYSLIB** file. The information consists of the information in this manual, except for sections on system loading and dumping.

4.2.25. @IDUMP — Interactive Dump Inspect

The @IDUMP utility program is the DCP/OS interactive dump analyzer.

The dump is analyzed with the same set of commands as DCP/OS debug mode (where applicable). For more information on debug mode, see the *DCP/OS Programming Reference Manual* (7431 6894).

Format

```
@IDUMP[,options] [qual*]file.elt
```

Parameters

options

H displays help information.

qual is the qualifier of the file name.

file is the file name.

elt is the name of the element.

Example

```
@IDUMP MYQUAL*DUMP.
```

!6. @IF — Conditional Processor

The @IF utility program is used with the @ELSE and @ENDIF utilities to set runstreams where sections of the runstream are executed conditionally. (See the @ADD and @START commands.) The @IF utility program checks the condition and skips a section of the runstream if the condition is not satisfied.

The following formats for the @IF command are possible (examples of which follow the description):

Format 1 `@IF,F[options] [[qual*]file[.elt]][,file[.elt...]]`

Tests for the existence of any of the files or elements listed. The element types (ABS, OMN, REL, SYM) must be specified as options. If no file or element is specified, the result is the same as though no item in the list were found.

Format 2 `@IF,C[options] [cond][,nnnn]`

Tests a hexadecimal value *nnnn* against the run condition word (see the @RCW command) according to the condition specified. If no condition is specified, EQ is assumed. If no value is specified, 0 is assumed.

Format 3 `@IF,D[options] [cond] [,nnnn]`

Tests the value *nnnn* against the processor type according to the condition specified. Values of *nnnn* should reflect the DCP processor types (5, 15, 25, 30, 35, 40, 50, 55, 600 Series, etc.). If no condition is specified, EQ is assumed. If no value is specified, 0 is assumed.

Format 4 `@IF,M[options] [cond] [,nnnn]`

Tests the value *nnnn* against the memory size of the DCP according to the condition specified. The value of *nnnn* is in 64K units (i.e., 2=128K, etc.). If no condition is specified, EQ is assumed. If no value is specified, 0 is assumed.

Format 5 `@IF,U[options] [cond] [,nnnn]`

Tests the value *nnnn* against the user boot keys in the boot keys (see option B) according to the condition specified. If no condition is specified, EQ is assumed. If no value is specified, 0 is assumed.

Note: *The user boot keys currently consist of 4 bits in the order 4,3,2,1*

Format 6 `@IF,B[options] [nnnn] [nnnn...]`

Tests the boot keys specified as *nnnn* in hexadecimal. If the logical AND of the given keys (*nnnn*) with the current boot keys is not 0, the test is successful. The boot keys are set up by the @MONFIG utility program. A typical use of this option is to check whether one of the user-reserved boot keys is set.

Format 7 `@IF[options] [nnnn][,nnnn...]`

Tests the exit code of the last program against the hexadecimal values in the list. If no values are given, 0 is assumed. The exit code is set as follows:

0	means clean termination.
DCP/OS error code	is a DCP/OS error.
User program abort code (ERR$)	

means program abort.

Parameters

options

H — is print help information and exit option.

L — is long display option.

N — is the negate (NOT) option. Reverses the action of the condition and processes the next statement if the conditions are not met.

A,O,R,S — is absolute, omnibus relocatable, or symbolic element types, respectively.

cond — is a condition used for object testing. This condition is one of the following:

EQ	object is equal to *nnnn*.
NE	object is not equal to *nnnn*.
GT	object is greater than *nnnn*.
GE	object is greater than or equal to *nnnn*.
LT	object is less than *nnnn*.
LE	object is less than or equal to *nnnn*.
ON	object AND *nnnn* is nonzero.
OFF	object AND *nnnn* is zero.

If no condition is specified, EQ is assumed.

Sample Runstream

A runstream using @IF might appear as follows:

```
@IF[,options] condition

    is Section A of the runstream

@ELSE

    is Section B of the runstream

ENDIF
```

If the condition is satisfied, then Section A of the runstream is executed while Section B is skipped. If the condition is not satisfied, Section A is skipped and Section B is executed. It is not necessary to have an @ELSE statement. The @ENDIF statement indicates the end of the conditionally executed section of the runstream. If no @ENDIF statement is paired with the @IF statement, @IF assumes that Section B extends to the end of the runstream.

Note that it is possible to nest IF/ENDIF structures. This means that Section A or Section B may contain an @IF condition and @ENDIF (and even @ELSE); these conditions are ignored if the section is skipped. It is your responsibility to make sure that each @IF statement is paired with an @ENDIF statement when nesting IF/ENDIF structures.

Example 1

```
@IF,F       FILE1.
@DELETE     FILE1.
@ENDIF
@CAT        FILE1.,///300
```

This runstream deletes **FILE1** if it exists and then catalogs a new **FILE1**. This runstream may be useful if you need to enlarge a file.

Example 2

```
@IF,B       0001
@START      TEST1
@ELSE
@START      TEST2
@ENDIF
```

This runstream starts a different batch run depending on whether or not the rightmost user boot key is set.

Example 3

```
@IF,C       GT,3FFF
@START      DEVNET
@ENDIF
```

The batch run DEVNET is started only if the run condition word (RCW) is greater than X'3FFF'.

Example 4

```
@PROGRAM1
@IF
@PROGRAM2
@ENDIF
```

This runstream starts PROGRAM2 only if PROGRAM1 completes without errors.

Example 5

```
@IF,D  GE,50
@PROGRAM5
@ENDIF
```

This runstream starts PROGRAM5 only if executed on a DCP with a processor type of 50 or more (i.e., a DCP/50 or 55).

Example 6

```
@IF,M  LE,16
@PROGRAM6
@ENDIF
```

This runstream starts PROGRAM6 only if the DCP has 1 MB or less of memory.

Example 7

```
@IF,U  EQ,0A
@PROGRAM7
@ENDIF
```

This runstream starts PROGRAM7 only if the contents of the user boot keys contain the value 0A.

4.2.27. @IOL — Disk I/O Error Log Analyzer

The @IOL utility program displays an edited printout of the I/O errors reported by the DCP/OS disk handler. Table 4–9 lists the available commands.

Table 4–9. @IOL Internal Commands

Command	Description	Command	Description
nnn	Display entry number nnn	RLU	Display next entry
RLD	Display previous entry	SRC	Display next entry for device
RPT	Repaint command line	HLP	Print this information
LMS	Interpret line module status	DVS	Interpret device status
SPS	Interpret supplementary status	EXI	Exit from program

Format

```
@IOL
```

Example

```
@IOL
```

Response

```
▸      ▸RLU  ▸RLD  ▸SRC  ▸RPT  ▸HLP  ▸LMS  ▸DVS  ▸SPS  ▸EXI          IOL 5R3.01
-------------------------  Fri 30 Oct 92   13:52:05  -------------------------
Entry Date    Time      Device   Error Func. Sector Byte  Line   Device Suppl.
      yymmdd hhmmss.ms           code        number count status status status
  4   930101 003156.583 SW2      8582  READ  0000E9 0000  0020   0000   0011



--------------------------------------------------------------------------------
 Error log size = 15  Current entry =  4  Number of errors =     4
--------------------------------------------------------------------------------
```

041DG100.DRW

Figure 4–1. @IOL Command Output

4.2.28. @LIMODUMP — Line Module Dump

The @LIMODUMP utility program produces a dump of the memory contents of a line module (LM) into a dump file. This file is produced in the same format as the contents dumped from the line module. Optionally, @LIMODUMP can simultaneously create an ASCII dump (printable element) of the same LM dump file. It can also process a dump file previously created by @LIMODUMP, then create a corresponding ASCII dump printable element. The printable element can then be transferred to a host system and printed.

Format

```
@LIMODUMP[,options] [qual*]dumpfile.[,[qual*]printfile.elt]
```

options

none creates a dump file only.

A displays all active LM dump files beginning with the dump file's name (for example, if the dump file's name is **LMDUMP**, then **LMDUMP, LMDUMP01, LMDUMP02** ... **LMDUMP09** will be displayed, but only if they exist and contain an active line module dump).

B bypasses the internal LIMODUMP time limit (4 minutes) allowed to dump a line module. This option should be used in conjunction with the L option to dump line modules with large memory capacities.

C compresses a repeated line of zeroes of an ASCII dump (print element) to significantly decrease the size of the print file. Use in conjunction with the I or L option.

D deletes the dump file header of an active line module and terminates the program. The file can then be used to create another line module dump.

F cycles the dump file if the specified file has an active line module dump.

H displays the HELP menu and ignores all other options and parameters that may have been entered.

I requires a previously created **LIMODUMP** dump file to be specified as input. The ASCII print file must be specified (ASCII dump output file).

L creates a print file and a dump file simultaneously. Both files must be specified on the call line. Use of this option adds significantly to the line module dump time; the B option should probably be used.

R allows you to specify a range in which to dump a line module. Begin the range by setting the starting address, then specify the number of bytes that you want dumped. If you are performing a full memory dump, this option is not necessary. This option can be used only with programmable line modules.

S displays on the screen the header block information of the dump file. Can be used with A or D options.

T displays the first eight words of each transmitted block of line module data, plus a block number and the time.

Note:

1. *The line module being dumped must not be assigned to a run.*
2. *If you do not catalog your own dump file or ASCII print file, LIMODUMP catalogs the file. It may not, however, catalog sufficient space to contain a full memory line module dump for some line modules. A LIMODUMP ASCII print file is cataloged with 300 blocks. A LIMODUMP raw dump file is cataloged with 2100 blocks.*

Parameters

*qual*dumpfile*	is the file name of the file to which the line module memory contents are to be sent, or a file containing a previous line module dump to be processed to create an ASCII dump. The file must be precataloged.
*qual*printfile.elt*	this parameter is optional, but you must supply it when you use the I or L options.

Note: *Qual is the qualifier of the file name. Default conventions apply. Refer to the @QUAL command in this section.*

Example of Performing a Raw Dump

Figure 4–2 illustrates an example of performing a raw dump.

```
▶@CAT rawdump.
▶FUP 5R3.01   Fri 16 Oct 1992 14:00:00
▶END CAT
▶@LIMODUMP RAWDUMP.
▶LIMODUMP   5R3.01  (RSYS-7R2.00F)  - Fri 16 Oct 1992 14:00:17
▶Enter in a Port number (0 - FF) or EXIT to end -.04
▶
▶14:00:41  Started LM dump of port 004....
▶
▶14:00:42  LM dump of port 004.  Completed - no errors.
▶END  LIMODUMP

▶@PRT,F rawdump
▶File:  Q$Q$Q$*RAWDUMP          Highest block:    17 (    5Kb)
 Modes: PUBLIC                      Opened by: -
 Vol:   SW0     Type: 8441-W             Size:   128 blocks (   32Kb)
 Last Updated:  14:00:00 on Fri  16 Oct 1992
▶
```

018DG100.DRW

Figure 4–2. Raw Dump Example

Example of Performing an ASCII Dump

The raw dump in this input file (from the previous example) is shown in Figure 4–3.

Note: *Only the first few lines of the ASCII dump printout are shown below. The leading number on each line is a file-relative hexadecimal byte address that does not necessarily map to the line module memory.*

```
▶@CAT PRNTFILE.,///1400
▶FUP 5R3.01   Fri 16 Oct 1992 16:01:21
▶END CAT
▶@LIMODUMP,I RAWDUMP.,PRNTFILE.DUMP
▶LIMODUMP   5R3.01  (RSYS-7R2.00F)  - Fri 01 Jan 1993 04:35:56
▶WARNING - Ascidump size of 1400 blocks may be insufficient.
▶
▶04:35:57  Started writing an ASCII dump file with the input rawdump file....
▶
▶04:36:02  Writing an ASCII dump. Completed - no errors.  ASCII dump written.
▶Approximately  48 ascii dump pages were written.
▶
▶END  LIMODUMP

▶@PRT,F PRNTFILE.DUMP
▶Spec  1 - Element name illegal
▶@PRT,T PRNTFILE.DUMP
▶Q$Q$Q$*PRNTFILE
▶SYM   DUMP
▶@PRT,F PRNTFILE
▶File:  Q$Q$Q$*PRNTFILE        Highest block:    86 (   22Kb)
 Modes: PUBLIC                      Opened by: -
 Vol:   SWO     Type: 8441-W              Size:  1400 blocks (  350Kb)
 Last Updated:  04:36:00 on Fri  01 Jan 1993
▶
```

019DG100.DRW

Figure 4–3. ASCII Dump Example

Example of Displaying an ASCII Dump File

Figure 4–4 illustrates an example of the output of an ASCII dump file.

```
@PRT,S PRNTFILE.DUMP
Q$Q$Q$*PRNTFILE.DUMP
  1.  LIMODUMP   5R3.01 - Fri 16 Oct 1992 14:04:05 (Created using option I)
  2.
  3.  ********************************************************************************
  4.  *  L I N E  M O D U L E  D U M P                                               *
  5.  *                                                                              *
  6.  ********************************************************************************
  7.
  8.   Port number:   n/a
  9.   Hardware id:   n/a
  10.  Microcode id:  n/a
  11.
  12. ********************************************************************************
  13. 00000  00 01 02 03  04 05 06 07  FF 13 0A 0B  00 04 00 14  ............
  14. 01060  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  ............
  15. 01070  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  ............
  16. 01080  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  ............
  17. 01090  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  ............
  18. 010A0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  ............
```

020DG100.DRW

Figure 4–4. ASCII Dump File Display

4.2.29. @LOG — Log Message

The @LOG utility program allows you to log messages in the system log file (**SYS$*SYSLOG**).

Format

```
@LOG ['message text['']
```

Parameters

message text	is the string to be logged. The quotation marks are necessary around the text only if the message string contains blanks.

Examples

```
@LOG  'the system behavior statistics are being collected'

@LOG  'Telcon restarted'
```

30. @MODE — Set Disk Type

The @MODE utility program sets the disk type for all subsequent @CAT requests until the next @MODE request. The specified disk type overrides the type specified on any @CAT requests. The @MODE utility is part of the @FUP program.

Format

```
@MODE[,options] type
```

Parameters

options

H	displays help information.
R	reverts to any disk type (i.e., type D).

type is the mass storage type (see the @CAT command for a list of valid types). If no type is given, @MODE displays the current disk type setting.

4.2.31. @MONFIG — DCP/OS Configuration Program

The @MONFIG utility program allows you to inspect, create, or change items in the physical device table (PDT) or the load path table (LPT). Refer to Section 2 for definitions of these system tables. This program leads you through the inspect, create, and change functions, which are accessed through a menu.

The initial PDT and LPT are taken from the file **SYS$*SYSCFG** on the bootstrap disk when the DCP is booted. During normal operation, these tables are kept in memory without changing the original tables on disk. The @MONFIG command allows you to inspect or change the online tables in memory (M option), copy the online tables to a file (W option), and update or create a new configuration file (U and I options).

Format

```
@MONFIG[,options] [[ qual*]file.]
```

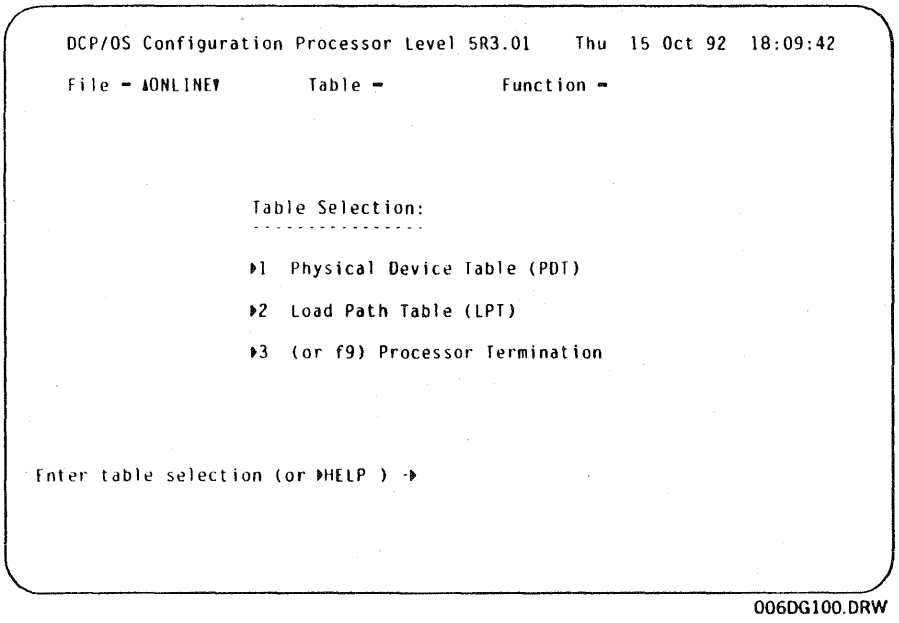
Parameters

options

B	suppresses full-screen mode for batch processing.
H	displays help information.
I	creates a new PDT/LPT (taken from the online tables) in the specified file.
M	updates the PDT and LPT configuration in memory with the option of copying it to a file. This is the default if no option is entered.
U	updates the PDT/LPT of the specified file without affecting the online tables.
W	copies the online PDT and LPT configuration to the specified file.

*qual*file* refers to the input file for the U option or to the new configuration file for the I option. The file name is required for the I and U options; it is not allowed for the M and W options. The M and W options will prompt you for a file name for the modified tables (the default file name is **SYS$*SYSCFG**). Default conventions apply for the file name qualifier. Refer to the @QUAL command.

Figure 4–5 shows the @MONFIG general menu screen.



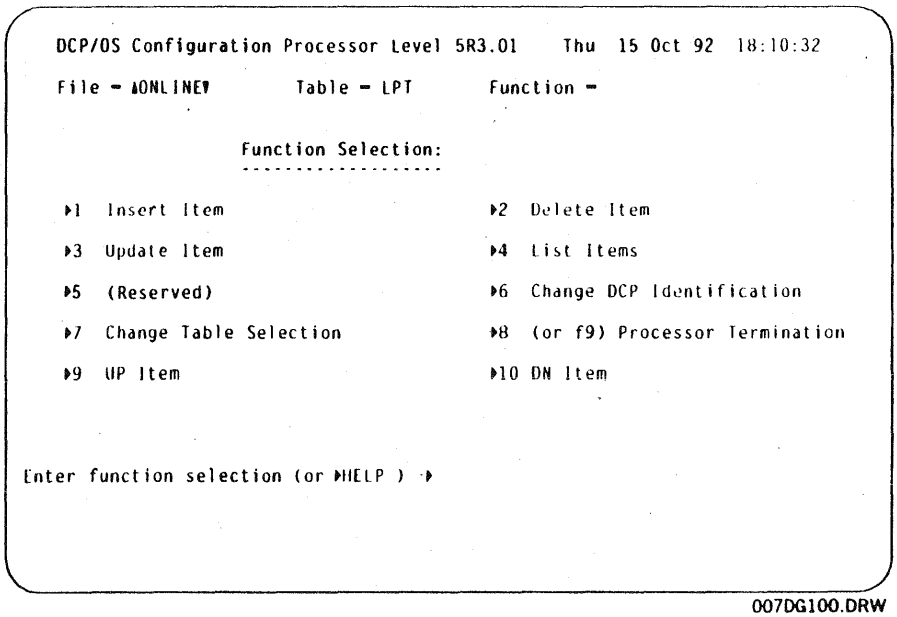
006DG100.DRW

Figure 4–5. General Menu Screen

The physical device table (PDT) contains the name and device-dependent details of every device configured for DCP/OS usage. This does not mean that every port on the DCP has to be configured in the PDT. Configure only the devices required for DCP/OS usage. Application programs may use any port not currently assigned to DCP/OS or to another run.

Devices in the PDT may be configured as initially UP or DOWN with the @MONFIG utility program (see Section 4). If initially marked UP, these devices are brought into an operational state at boot time. With the exception of the system volume, every device in the PDT may be dynamically brought up or down at run time. If the load path indicates that the DCP/OS is not memory-resident, the volume (disk drive) from which the DCP/OS is loaded at boot time is tagged as the system volume.

Figure 4–6 shows PDT function selections.



007DG100.DRW

Figure 4–6. PDT Functions

Table 4–10 describes the physical device table functions.

Table 4–10. Physical Device Table Functions

Insert item	@MONFIG selects an unused entry in the PDT and allows you to set up all fields in the PDT entry using menu screens for each field.
Delete item	@MONFIG allows you to specify the PDT item by name or number. A page of PDT items is displayed to help you select an item. You can scroll through the items if there is more than one full screen.
Update Item	@MONFIG allows you to specify the PDT item by number or by name. A page of PDT items is displayed to help you select an item. You can scroll through the items if there is more than one full screen. When you choose the PDT item, @MONFIG allows you to modify any fields in that item by using menu screens.
UP Item	@MONFIG allows you to specify the PDT item by number or by name. A page of PDT items is displayed to help you select an item. You can scroll through the items if there is more than one full screen. When you choose the PDT item, @MONFIG causes the item selected to be marked as UP.
DN Item	@MONFIG allows you to specify the PDT item by number or by name. A page of PDT items is displayed to help you select an item. You can scroll through the items if there is more than one full screen. When you choose the PDT item, @MONFIG causes the item selected to be marked as DN.
List items	@MONFIG displays a screenful of PDT items, then allows you to scroll through the remainder of the PDT items one page at a time.
Change DCP identifier	@MONFIG allows you to change the DCP identifier. This is the identifier that appears in the banner on the console mode workstations and is analyzed in formatted dumps.

Upon installation of the DCP/OS, the PDT is designed to use the first terminal that polls back to the system. You can assign a specific port number for the DCP/OS workstation by changing the PDT. You can use the PDT to modify or add a channel, set port numbers, and so on. For example, you may want to configure a channel so you can use DCPFT to transfer files.

Figure 4–7 shows LPT function selections.

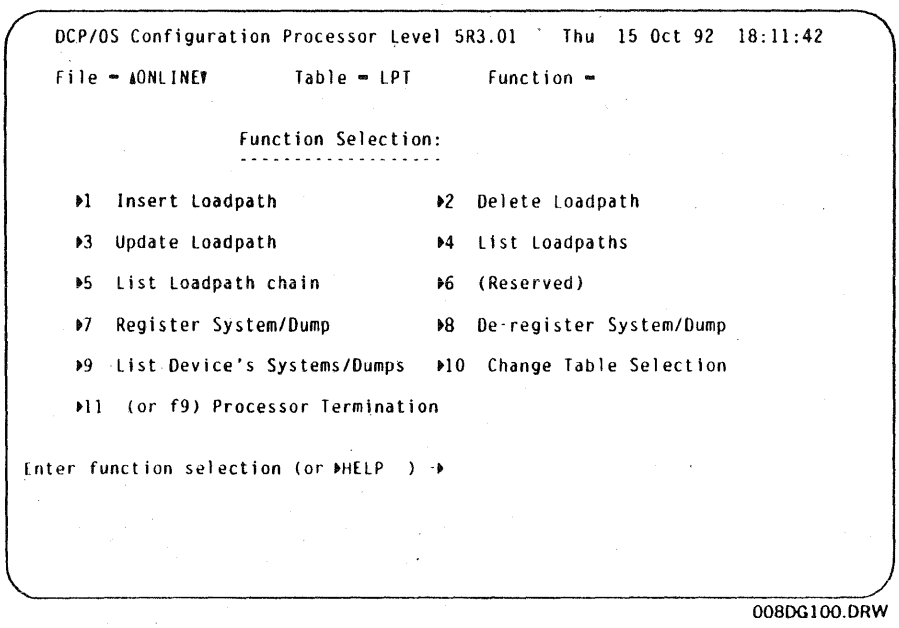


Figure 4–7. LPT Function Selections

Table 4–11 describes the load path table functions.

Table 4–11. Load Path Table Functions

Insert item	You can select a free item or you can let @MONFIG choose one for you. Once the item has been chosen, @MONFIG allows you to modify or initialize each field in the LPT item using a menu screen.
Delete item	@MONFIG allows you to specify the item number. A page of LPT items is displayed to help you choose the item number. You can scroll the display if there is more than one page.
Update item	@MONFIG allows you to specify the item number. A page of LPT items is displayed to help you choose the number. You can scroll through the items if there is more than one full screen. When the LPT item has been chosen, @MONFIG allows you to modify any fields in the item using menu screens.
List items	@MONFIG displays a screenful of LPT items, then allows you to scroll through the remainder of the LPT items one page at a time.
List chain items	@MONFIG displays a chain of LPT items starting from a specified LPT. The display shows all fields associated with this item, including successor items and alternate items. A successor item is acted upon after successful completion of the previous item. An alternate item is acted upon after unsuccessful completion of the previous item.

continued

Table 4–11. Load Path Table Functions (cont.)

Register system/dump	@MONFIG allows you to specify the file or element containing a DCP system or dump file to be registered. It then requests that you choose the system number to be used for the registration. To help you choose, a list is displayed of all the systems currently registered on the device containing the system to be registered. This list displays the information contained in the load header of each system: file name, element name, time and date of build, and level information provided at build time. Note that the file and element name in the load header are written there by the @MONFIG register system function, not by the build program. Thus, they accurately reflect the current situation and not the name of the file used for the build. @MONFIG then updates the appropriate system list entry with a pointer to the newly registered system and writes the updated system list to the device. You must be privileged to register a system or dump file.
Deregister system/dump	@MONFIG allows you to specify the device containing the system to be deregistered. @MONFIG then clears out the pointer to the system contained in the selected system entry and writes the updated system list to the device.
List system/dump on a device	@MONFIG allows you to specify a device.

Caution

If you configure the load path to perform a local dump followed by a DOWNLOAD, your dump files are deleted.

@MONFIG allows you to set or clear any of the boot keys for a specific load path. Figure 4–8 shows the screen display of the Update Path option of the Load Path Table menu item.

```
DCP/OS Configuration Processor Level 5R3.01    Thu  15 Oct 92  18:14:46

File = ▲ONLINE▼        Table = LPT        Function = Update Path

  LOAD-                                                --CHAIN LIST--
  PATH #   TYPE     DEVICE    FILE #    PORT #    ALT #    SUC #    KEYS
    01     LOAD     SW0         00                 00               B000
                                                                      ^
Below is a list of Boot keys, their current state and descriptions.
To change the setting of any key, enter the key number, and XMIT.

 KEY   STATE  DESCRIPTION              KEY   STATE  DESCRIPTION
 ▶13    ON    Auto Re-boot             ▶12    ON    Auto Start-up
 ▶11    OFF   Release Bootstrap Mem.   ▶10    OFF   Hold Batch Runs
 ▶9     OFF   Hold Demand Runs         ▶8     OFF   Cache Disable
 ▶7     OFF   Force Memory-Resident    ▶6     OFF   Hard Clear All LMs
 ▶5     OFF   Force Download Format    ▶4     OFF   Force Download Purge
 ▶3     OFF   User Boot key 3          ▶2     OFF   User Boot key 2
 ▶1     OFF   User Boot key 1          ▶0     OFF   User Boot key 0

XMIT or enter a key number (or ▶HELP  ) -▶
```

009DG100.DRW

Figure 4–8. Default Values of Boot Keys

Table 4–12 describes the boot keys you can set or clear for a specific load path.

Table 4–12. Boot Keys

Key	Name	Description
13	Auto Reboot	Allows system to reboot automatically after a fatal DCP/OS error.
12	Auto Startup	Enables the auto startup feature.
11	Release Bootstrap Mem.	Returns memory used by the bootstrap to available memory pools. CAUTION: A system dump will be useless if the bootstrap memory is released. This is because the system software that dumps memory to a file uses the bootstrap area and will overwrite any data located there.
10	Hold Batch Runs	Disallows ST and @START commands.
9	Hold Demand Runs	Disallows @RUN commands.
8	Cache Disable	Disables cache; the normal mode is to have caching active. For a description of caching, see the section 6, *Memory Management*
7	Force Memory-Resident	Most DCP/OS segments are kept on disk and copied into memory only as needed. When this option is on, the entire DCP/OS will be kept in memory at all times.

continued

Table 4–12. Boot Keys (cont.)

Key	Name	Description
6	Hard Clear LMs	Hard clear all of the line modules (LMs). This disconnects LM buses from both partitions and LMs are no longer loaded.
5	Force Download Format	Used as a development tool.
	Force Download Purge	Causes the file space (on disk(s) specified on the @DOWNLOAD command) to be marked as unused, thus allowing the whole disk to be reused
3	User Boot Key 3	Can be retrieved by software through the LINFO$ service. For more information, see the *DCP/OS Programming Reference Manual* (7431 6894).
2	User Boot Key 2	Can be retrieved by software through the LINFO$ service. For more information, see the *DCP/OS Programming Reference Manual* (7431 6894).
1	User Boot Key 1	Can be retrieved by software through the LINFO$ service. For more information, see the *DCP/OS Programming Reference Manual.*
0	User Boot Key 0	Can be retrieved by software through the LINFO$ service. For more information, see the *DCP/OS Programming Reference Manual.*

Auto reboot

When the DCP/OS encounters a fatal system error (see Table A-11), the auto reboot option in the load path table indicates to the DCP/OS whether you must reboot the system (when this option is on) or stop. The current state of this option comes from the load path specified by the load switches value at the previous boot (not the current value of the switches) and is assumed to be on for port path boots. The default value of this option, if not manually changed, is on. Auto reboot is typically be turned off only by system personnel for troubleshooting.

Auto startup

When the auto startup option is set to on (the usual setting) the runstream in **SYS$*SYSJOB.STARTUP** is automatically started after the DCP/OS has been initialized. The auto startup feature can be changed only for load path boots from local disk. Local loads through port path always start the **SYS$*SYSJOB.STARTUP** runstream. Host loads invoke the **SYS$*SYSJOBx.STARTDOWN** runstream whether the auto startup option is set to on or not.

Force memory resident

When the DCP/OS is downloaded from the host it is copied to the system volume. Various portions of the DCP/OS are used so infrequently that they are normally erased from DCP memory when that memory is needed by a user program; they are loaded back into memory from the system volume only when they are required.

The force memory resident option in the load path table prevents this from happening by not allowing any of the DCP/OS to be erased from DCP memory. This can improve execution speed, as long as the DCP has enough memory available to meet the needs of all active runs.

Another useful aspect of having a memory resident DCP/OS is that none of the system disks is then marked as the system volume. Many disk operations, such as disk formatting, require that the disk be down (see Section 3 for the DN command) and that the system volume not be down. A memory-resident DCP/OS can also avoid dependence upon an unreliable disk.

Load path 0 always forces DCP/OS to run memory resident for those DCPs that can be loaded from the boot diskette.

To verify that the DCP/OS is not memory resident, use the following command:

```
@@CONS RD *MON*
```

In a memory-resident system, the number of transient segments of code and data should be 0 for the run-name *MON* (the DCP/OS). In general, the number of transient dynamic segments is not 0.

Force Download Disk Format/Purge

The format/purge boot keys, specified on a @DOWNLOAD load path, take precedence over the @DOWNLOAD device directives. The @DOWNLOAD device directives only apply when the format/purge boot keys are marked as off.

4.2.32. @MSG — Send Message

The @MSG utility program allows you to send messages to other DCP/OS users on the system.

Format

```
@MSG,[options] ['']message text['']
```

Parameters

options

C	sends message to console mode workstations.
H	is the display help information option.
L	is the display banners option.
W	is the wait for input after display (does not wait in batch, ignored if C option).

message text is the string to be printed.

Note: *Surrounding quotation marks ['] are required only if the message string contains blanks.*

Examples

```
@MSG,C  'File purging at 17:00'
@MSG,W  'Hit XMIT to continue'
```

33. @PACK — Pack Program File

The @PACK utility program resides within the FUP. It removes deleted elements from a file and packs the remaining elements to the front of the file. This makes space available at the end of the file so that new elements can be added.

Format

```
@PACK[,options] [qual*]file.
```

Parameters

options

H	displays help information.
L	lists the element names as they are processed.
-	no option (does not list the element names).

qual is the qualifier of *file*. Default conventions apply. See the @QUAL command.

file is the name of the file to be packed.

Example

```
@PACK MYQUAL*FILE.
```

4.2.34. @PATCH — Patch Absolute Element

The @PATCH utility program applies patches to an absolute element or file, that is, one containing a program. A patch location may be verified before applying a patch and locations may be referenced by offsets from symbolic base addresses (that is, segment name + offset).

The patches may be applied either directly to a program (K option) or may be applied by creating a new output element using an intermediate work file.

The target program dictionary is used for symbolic references. @PATCH also uses an internal dictionary of useful offsets within the absolute element.

All numeric input is assumed to be decimal. If input is preceded by a 0, it is assumed to be hexadecimal.

The third parameter on the program call line, which is optional, creates a listing element containing the complete dialog.

The @PATCH program uses internal commands to apply the patches to the element or file. These commands are listed in Table 4–13.

If a major error occurs, the @PATCH program is terminated immediately.

Table 4–13. Internal @PATCH Commands

Command	Explanation
COM	Inserts a command in listing
END	Designates end of input
LEV	Checks current and ABS revision levels
PRT	Prints out specified location contents
REP	Replaces current contents of location witha value
SRC	Checks original @BUILD qual*file.element and date
VER	Verifies contents of specified location

Format

```
@PATCH,opt  [iqual*]ifile[.ielt],[[oqual*]ofile[.oelt]],
 [[lqual*]lfile.[lelt]]
```

Parameters

options

A updates an active file.

H displays help information.

I is file to element or element to file.

K applies patches directly.

Caution

In this mode, you cannot undo the patches by terminating the @PATCH program because patches are applied as they are entered.

W	indicates that the work file (P$WRKFLN) is not to be deleted on termination.
iqual	is the qualifier of *ifile*.
ifile	is the input filename.
ielt	is the input element name.
oqual	is the qualifier of *ofile*.
ofile	is the output filename.
oelt	is the output element name.
lqual	is the qualifier of *lfile*.
lfile	is the listing file name.
lelt	is the listing element name.

4.2.34.1. COM

The COM command inserts a comment in the output listing.

Format

```
COM abc...
```

4.2.34.2. END

The END command signals the end of input.

Format

```
END
```

34.3. LEV

The LEV command checks the current patch version and patch revision levels of the ABS program and optionally checks the @BUILD LEV value. If no parameters are entered, the current level is displayed.

***Note:** If patches have not been applied, 0.0 is the level number.*

Format

```
LEV ver.rev[,'lev-string']
```

Parameters

ver.rev is the current @PATCH level.

ev-string is the string entered in the @BUILD LEV command.

34.4. PRT

The PRT command prints out the contents of the specified locations.

Format

```
PRT name[+offset] [wrdcnt]
```

Parameters

name is the segment name as defined in SSTDEF.

offset is the number of 16-bit words.

wrdcnt is the number of words to display (default is 8).

4.2.34.5. REP

The REP command replaces the current contents of the specified location with the specified value.

Format

```
REP name[+offset] data
```

Parameters

name	is the segment name as defined in SSTDEF.
offset	is the number of 16-bit words.
data	is the new value.

4.2.34.6. SRC

The SRC command checks the output file or element and date of @BUILD or @DCPBUILD. If no parameters are entered, the file or element name and date are displayed.

Format

```
SRC bfile.belt[,day,month,year]
```

Parameters

bfile	is the build file name.
belt	is the build element name.
day,month,year	is the build date.

34.7. VER

The VER command verifies the contents of the specified location. If verification fails, any REP commands will not take effect.

Format

```
VER name[+offset] data
```

Parameters

name	is the segment name.
offset	is the number of 16-bit words.
data	is the current value.

Example

The following patches the program MON*LES.EXAM and creates a listing element.

```
@PATCH MON*LES.EXAM,TEMP.EXAM,*PLIST.EXAM

PRT EXAM+010 32          prints a few words.
VER EXAM+012 0B801       verifies a location.
REP EXAM+012 0B802       replaces a location.
END
```

4.2.35. @PORT — Display I/O Port Information

The @PORT utility program provides hardware and software information for line modules and IOP statistics.

Format

@PORT,*options*

Parameters

options

A	attempts to access all the line modules.
C	prints details of line modules in the communications line controller (CLC) from the requested port.
H	displays help information.
I	interrogates the line module rather than relying on previous information.
L	loads the line module if the status was successfully obtained. After the print number and the status of port is displayed, the following message appears: `Enter microcode ID of file.element name to be loaded into the line module:`
M	displays commonly used line module code identifiers.

Note: *Options* A, C, *and* L *are mutually exclusive.*

To exit the @PORT program, enter EXIT.

Example

```
@PORT,I
5
EXIT
```

checks port 5 status.

4.2.36. @PRIV — Establish Privileged Mode

The @PRIV demand mode command establishes privileged mode. This allows you to execute privileged programs and to modify real memory in debug mode. A password is always required on the @PRIV command.

The @PRIV command is also used to change the password. Do not use the default password (MONITOR), since it is always used after the system is rebooted. However, the default password (MONITOR) must be used the first time, as it is in the example.

Format

```
@PRIV password[,newpassword]
```

Parameters

password	is the password; eight-character maximum.
newpassword	is the new password; eight-character maximum.

Example

```
@PRIV PASSWORD
```

Response

```
▸ READY
```

Note: ▸ READY *is displayed only after a successful change to privileged mode.*

37. @PRT — Display Print File or Element Information

The @PRT demand mode command displays file or element information and can be used to list a symbolic element. The PRT command without any parameters or options displays a list of all online files, preceded by a heading with the volume name. The name of any file that is in a logical down state is enclosed in parentheses. (See @CAT for information pertaining to files in an up or down state.) Temporary files are denoted by a leading hyphen (-).

Format 1

```
@PRT[,options] [qual*]file.[elt][,n]
```

Format 2

```
@PRT,D volumename
```

Parameters

options

A	prints the absolute elements in file (requires T option).
B	prints the table of contents backwards.
D	displays files on given volume.
F	displays file information, including size.
I	displays temporary files created by the run.
L	provides a long list of element names (with T option).
N	inhibits line numbering.
O	prints the omnibus element (requires the T option).
R	prints the relocatable element (requires the T option).
S	prints the symbolic element.

T	lists elements.
X	ignores errors.
file	is the file being displayed
qual	is the qualifier of the specific file being displayed
elt	is the name of the element within the file being displayed
n	is the number of elements to display (with T option).
volumename	is the name of any online volume (used only with @PRT,D).

Examples

@PRT	lists the names of all files.
@PRT,TL MYQUAL*FILE	lists the table of contents of file **MYQUAL*FILE**.
@PRT,S MYQUAL*FILE.LOOPJO	lists the text of symbolic element LOOPJOB in file **MYQUAL*FILE**.
@PRT,D SW0	lists the names of all files on volume SW0.
@PRT,TLB **SYS$*SYSLIB.,5**	list the last five elements backwards of file **SYS$*SYSLIB**.

18. @QUAL — Set Assumed Qualifier

The @QUAL utility program resides within the FUP. It establishes, resets, or displays the current assumed qualifier.

Format 1 @QUAL

Format 2 @QUAL,P

Format 3 @QUAL,R

Format 4 @QUAL *qual*

Parameters

options

no option	displays the current project-ID and the assumed qualifier.
P	changes the current project. The run must be privileged.
R	causes the assumed qualifier to revert to the project-ID. Any *qualifier* parameter is ignored.
qual	is a name from one to six characters. Valid characters are the same as for filenames. (See the @CAT command.) An invalid entry causes the assumed qualifier to revert to the project-ID, just as does the R option.

Example

```
@QUAL MON
```

Additional Discussion

If you specify the file name without any qualifier, for example: *file*, the file name defaults to *project-id*file*. In this case, the project-ID is the assumed qualifier.

If you specify the file name, for example: **file*, then the qualifier is implied like this: *assumedqual*file*. The implied qualifier is the qualifier name from the last @QUAL command. If no @QUAL command has been entered prior to the file name reference, the project-ID is used as the default qualifier.

39. @RCW — Display/Modify Run Control Word

The @RCW utility program displays the current run control word (RCW) and allows you to modify it.

Format

```
@RCW,[options] [value]
```

Parameters

options

A	performs logical AND operation using the current RCW and value. Replaces RCW with new result.
H	is help information.
L	prints the banner line.
O	performs logical OR operation using the current RCW and value. Replaces RCW with new result.
S	sets the RCW to *value*.
X	performs logical XOR operation using the current RCW and value. Replaces RCW with new result.

value is the hexadecimal value (maximum value is FFFF).

Example

```
@RCW,S 1234
```

4.2.40. @RUN — Initiate Run

The @RUN demand mode command initiates a run and must be the first statement in a batch run. When entered in console mode, it initiates a demand run.

If you want to run a program that solicits input, make sure you are not in @@CONS mode. (You terminate @@CONS mode with an @@END command.) If you run in console mode, the input is processed as a console command and is not be passed to the program.

In demand mode, the screen delete point is set to the top of screen, the screen is cleared, and a "message of the day" (held in **SYS$*SYSJOB.LOGON**) is displayed. If this element does not exist, no message is displayed.

Format

```
@RUN[,[options][/priority]] run-name[,,[project][,spool-size]]
```

Parameters

options

D	causes automatic debug entry on program errors. If an error occurs and D is not specified, an error message is displayed and the program is terminated.
N	inhibits the creation of a spool file for this batch job.
X	sets the workstation in console mode (@@CONS mode) while also allowing demand mode statements to be entered if they are preceded by the @ symbol. To run demand mode utilities requiring user input, @@END must be entered to terminate @@CONS mode. Console messages will be displayed in either mode unless a program is executing.

priority — a single alpha character that specifies the priority of the run. The highest priority is A and the lowest is Z. The run priority is used when the system is in a critically low buffers situation (hard throttle) to determine which jobs should be terminated (starting at the lowest priority).

run-name — is an alphanumeric name from 1 to 6 characters used to identify the run.

project — is an alphanumeric name from 1 to 6 characters describing the default qualifier that will be used on any file references. If no valid project is supplied, the QQQ$ is used as the default.

spool-size — the spool file size in blocks (default is 128 blocks). The spool file namr is **SYS$*P$*run-name***.

Example

```
@RUN,/M MARY,, SYS$
```
This example runs priority M.

Note: *The comma (,) immediately following @RUN is required in batch, but not in demand mode.*

4.2.41. @SPLIT — Split/Recombine File

The @SPLIT utility program allows you to break a file into multiple smaller files, and to recombine a set of split files into the original file. The purpose of the SPLIT program is to create a set of small files that can each be copied to a diskette. You can then take these diskettes to a different DCP and recombine the split files into the original file (typically a large program or data file).

When files are split, they are sequentially numbered, based upon the supplied file name parameter. An incremental number is added to the end of the file name for each split file that is created. Each additional split file is cataloged at the same size as the original output file (split0). The output file size splits from the first of the sequential split files.

The P option can be useful in recombining files from diskettes, allowing you to switch diskettes and bring the diskette drive up and down.

Format

`@SPLIT[,`*options*`]` `[`*qual1**`]`*oldfile*`.,[`*qual2**`]`*split0*.	(to split a file into several smaller files).
`@SPLIT,C[`*options*`]` `[`*qual2**`]`*split0*`.,[`*qual3**`]`*newfile*.	(to recombine split files into one file).

options:

C	combines SPLIT0., SPLIT1., SPLIT2., and so on, into NEWFILE.
H	accesses the help screen.
N	minimizes output.
O	unconditionally overwrites precataloged files.
P	pauses before opening each file.

V	allows SPLITN. to be cataloged in any volume (default is same).

Parameters

oldfile	is the original file that is to be split into smaller files.
qual	is the qualifier of the file to be split into smaller files.
split0	is the intermediate split files.
newfile	is the target file that is to be created from the split files.

4.2.42. @START — Start Batch Run

The @START demand mode command initiates a batch run. The runstream is taken from the specified element, which must start with an @RUN command. If the file name is not specified, the DCP/OS searches the files **SYS$*SYSJOB**, **SYS$*SYSJOBX**, **SYS$*USRJOB**, and **SYS$*USRJOBX** in that order.

Format

```
@START[,n] [qual*]file.element[,[RCW][,spool-size]]
```

Parameters

n	means no spool-file is created for this started batch job.
RCW	sets initial RCW for the run to the specified hexadecimal value (maximum value is FFFF).
spool-size	specifies the size of the spool-file created for this started batch job (default is 128 blocks). The spool file name is **SYS$*P$RUN0*n***, where *n* is the run number. This name is superseded if the @RUN command in the runstream specifies a spool file size (the name becomes **SYS$*P$[*run-name*]**) or an @BRKPT command is encountered in the runstream.
*qual*file.elt*	is the name of the element containing the runstream.

Example

```
@START MYQUAL*FILE.LOOPJOB
```

43. @SYS — Display System Status

The @SYS utility program displays information about the system or about jobs that are currently running.

Format

```
@SYS[,options] [,,file.[elt]]
```

Parameters

options

B	displays memory bank map.
D	displays debug system log file entries.
E	exits SYS.
F	displays system log file.
G	displays information related to system which includes the time and date that the operational system was booted, provided the D (date) console command was executed, and the load path details.
H	displays help information.
J	displays information related to current jobs.
L	displays system levels.
M	displays memory details.
Q	bypasses prompts when used with D option. All debug log entries are selected.
R	repeats the display every five seconds.
S	displays IOP statistics.

Note: *The IOP statistics feature is DCP model-dependent. If your particular model is not supported, a statistics unavailable message is displayed in response to the S option.*

T — displays system behavior statistics.

V — displays free space on disk volumes.

file.elt — is the optional file or element that is to hold the redirected listing away from the screen. This is used with the F or T option. The file or element must already exist.

Note: *When using the* R *option, the only way to exit is by using the @@X command.*

Example

```
@SYS,S
```

Additional Discussion

The @SYS utility provides the display of system log file entries and system behavior statistics (one of the log file entry types). These entries can be from either the system log file (**SYS$*SYSLOG.**) or its copy in a specified file.

When displayed from the active system log file, the file is actively changed by the system during the display session. Depending upon the frequency of file updates by the system and the duration of the display session, this can result in unusual log entry sequence displays.
To minimize these sequence displays, the system log file can be copied when it is not active. Displays from the copy of the log file are then requested. Multiple passes can be performed on the copy of the log file.

- **System Log File**

  The scanning for log entries to be displayed is limited by the start date/time, end date/time, log type and subtype, and run-ID. Provision is made to display detailed information along with the header information.

  Supported log types include user-specified types, system behavior statistics, @LOG JCL messages, file request packets for file operations (excluding read/write), PP state items, console messages, debug screens, and line module (LM) load blocks.

- **System Behavior Statistics**

  The scanning for statistical samples to be displayed is limited by the start date/time and end date/time.

  A sample is displayed through a multitude of screens. You can move between the displays in a sample sequentially or randomly, thereby allowing you to review a display that you have already viewed. When you are done viewing a sample, request the next sample. You can quickly scan sampling times by repeating next sample requests without any displays within a sample.

  Statistical samples provide information that assists in problem solving, determining average and peak loads, and system profiling.

- **Low Overhead Statistics**
  - PP statistics table
  - CP % utilitization
  - Dispatcher counts
  - IOP % utilization table
  - Memory utilization
- **High Overhead Statistics**
  - Individual initial procedure task timeslices
  - Transient segment statistics
  - Stack allocation failures

- Disk I/O statistics
- PP paired messages
- Task statistics

44. @TRCPP — Collect PP Traces

The @TRCPP program prompts you for a port number (which must be given in hexidecimal format), collects PP traces, and puts them in a trace file. To terminate TRCPP, use @EOF or function key 9 (F9).

Format

```
@TRCPP[,bihnptw] [trcfile.]
```

Parameters

options

b	batch mode (do not check for @EOF, with T)
i	stops output on every full screen (with P)
h	accesses the help page
n	minimizes output
p	prints contents of trace file (the printout is similar to PP traces in a DCP dump)
t	traces a selected PP
W	wraps around trace file when full (with T)
trcfile	file in which to collect PP traces (default file name is TRCFILE)

This program works with all port processor programs that use the TRACEDESC standard method of tracing. This excludes all channel port processors.

Example

```
@TRCPP,TPI
```

Response

```
▶@TRCPP,TPI
▶TRCPP   5R2.00  (RSYS-7R2.008)  - Fri 16 Oct 1992 11:08:50
▶- Which PP would you like to trace?  [ppid/ln]  ▶01C
▶- Traces of PP 01C are collected in file TRCFILE
▶- To terminate tracing, use @EOF or function key 9
▶
▶- Tracing was terminated
▶- Traces taken from file TRCFILE
▶
▶  BEGIN OF TRACE
▶  11:08:55.149                                  4002 0000 2EA0 0371 1004 8013
▶  11:08:55.233                                  4002 0000 2EA1 0171 1004 8013
▶  11:08:55.247 0000 0000 C002 01D1 0000 0000
▶  11:08:55.257 0000 0000 C002 03D1 0000 0000
▶  11:08:55.278                                  4002 0000 2EA2 0371 1004 8013
▶  11:08:55.362                                  4002 0000 2EA3 0171 1004 8013
▶  11:08:55.375 0000 0000 C002 01D1 0000 0000
▶  11:08:55.384 0000 0000 C002 03D1 0000 0000
▶  11:08:55.406                                  4002 0000 2EA4 0371 1004 8013
▶  11:08:55.491                                  4002 0000 2EA5 0171 1004 8013
▶  11:08:55.508 0000 0000 C002 01D1 0000 0000
▶  11:08:55.518 0000 0000 C002 03D1 0000 0000
▶END TRCPP
```

011DG100.DRW

Figure 4–9. TRCPP,TPI Ouput

15. @TUNER — Tune System Parameters

The @TUNER utility program allows you to tune (modify) and displaying of various system parameters. You must be privileged to use @TUNER.

Primary reasons for system tuning include:

- Minimum memory configured
- Statistics indicate excessive throttling/thrashing
- Traffic peaks force excessive bank break-up
- Unusual program mix (for example, production/test/debug)

The following are typical tuning parameters:

- Initial buffer pool size (MINBUFS)
- Transient sticking factor (ETPRTS and ETOSTS)
- Minimum cache banks (MINBCA)
- Soft throttle level (EWPERC or EWPBUF)
- Hard throttle time before terminating runs (60000)
- Logging on or off

These tuning areas usually satisfy most DCP and Telcon system requirements.

> **Caution**
>
> Approach dispatcher parameter tuning and other memory timers with caution.

Format

```
@TUNER[,options] [in],[out],[list]
```

Parameters

options

B	sets no interactive mode.
D	sets hexadecimal value display.
F	sets continuous display mode (must use for hard copy). Default is full screen mode with XMIT to continue.
H	displays help information.
L	lists system parameters.
V	lists system variables.

in — is the file from which to get the DCP/OS absolute system with the SYSINF system parameters. Normally, this would be **SYS$*DCPOS**. which is a data rather than a program file. See @MONFIG in the LPT section. If not specified, SYSINF in memory is the origin.

out — is the file to which the updated DCP/OS absolute system with the updated SYSINF system parameters will be written. Normally, this would be **SYS$*DCPOS**. See @MONFIG in the LPT section. If not specified, SYSINF in memory is the destination.

list — is the name of the optional file (and element) that is to hold the redirected listing away from the workstation.

Sample Runstream

```
@RUN TUNER,,SYS$

@PRT SYSJOB.TUNE

@PRIV Monitor

@TUNER,B ,,TUNER.
mp                    selects memory pool page.
```

`minbnk 100`	changes minimum banks to 100.
`mp`	redisplays memory pool page for listing.
`EXIT`	exits and updates.
`@FIN`	lists file **SYS$*TUNER**.

Examples

```
@TUNER,L
```

This processor call lists the system parameters.

```
@TUNER,V
```

This processor call lists the system variables, then prompts for modification of system parameters.

Additional Discussion

Processing is directed from the specification prompts.

Parameter keywords are as displayed on the screen, and can be entered either from the prompt line or by tabbing to the desired entry.

Direction keywords are shown in Table 4–14.

Table 4-14. Direction Keywords

Keywords	Direction
1 or mp	Display memory pool parameters
2 or mt	Display memory timer parameters
3 or cd	Display cache/disk management parameters
4 or rp	Display run/process management parameters
5 or ls	Display log/statistics management parameters
6 or ms	Miscellaneous parameters
7 or sv	Display system variables
8 or exit or e	Exit utility (will prompt for update)
h or help	Display the help screens
m or menu	Display menu (prompt for direction)
u or update	Update the destination

The @TUNER program allows you to modify an extensive collection of system parameters that affect system behavior. If these parameters are modified on the disk file containing the DCP/OS, they are in effect on the next local load of that particular DCP/OS. If these parameters are modified in the memory of the current operational DCP/OS, they are in effect on the subsequent access of the particular parameter by DCP/OS.

When @TUNER is used to modify whether the system log file is active or inactive, @TUNER requests the respective open or close of the system log file. When the log file is open and active, log entries can be made. The system log file is maintained in wraparound mode on a random file with the highest block written as the full size of the file.

When @TUNER is used to modify whether statistical samples are to be collected or whether the collection is complete, @TUNER performs the necessary actions to allow or terminate the collection. The event-driven statistics are maintained cumulatively. These statistics are not reinitialized for each current sample period, but are cumulative over the total sample period. The detailed statistics require higher overhead in their collection.

The following parameters are of particular importance when loading the system:

- **System Log File**

  By default, the system log file (**SYS$*SYSLOG.**) is activated either at system initialization or at download completion. If you need to bring down the disk on which the log file resides, you must use either the LOGO console command or the @TUNER utility program to deactivate the log file. You can then bring down the disk.

  The @TUNER utility program is used to specify whether the log file is active or inactive, where it resides, and how large it is.

  If these parameters are modified on the disk file containing the DCP/OS, these parameters are effective on the next local load of that DCP/OS.

- **System Behavior Statistics**

  By default, the system does not collect system behavior statistics. The @TUNER utility program is used to specify whether statistical samples are collected, how often the samples are taken, how many samples are taken, and which categories of samples are collected. If these parameters are modified on the disk file containing the DCP/OS, these parameters are effective on the next local load of that DCP/OS.

Memory Pool Management

Memory pool parameters are accessed by selecting Direction Keyword 1 or mp from the @TUNER main menu. Figure 4–10 approximates the resulting screen display.

```
TUNER  5R3.01  (RSYS-7R2.00B) - Thu 15 Oct 1992 18:17:02

                        (1) Memory Pool Management

Memory Pool Parameters                          Upper-range Current Key        New
-------------------------------------------------------------------------------

%age of 128b buffers used to enter soft throttle.... 100      80 ▸ewperc       80
Number of 128b buffers used to enter soft throt.. 131071  13027 ▸ewpbuf    13027
Maximum 128b buffer number allowed for soft throt 131071 131071 ▸maxewp   131071
Number of 128b buffers to leave soft throttle.... 131071     26 ▸arnbuf       26
Available 128b buffers to stop bank breakup...... 131071  32767 ▸maxbufc   32767
Minimum number of 128b buffers in the system..... 131071   1152 ▸minbufs    1152
Maximum number of 4Kb segments in the system..... 4096      8 ▸maxsegs       8
Minimum number of 16Kb banks in the system....... 1024     48 ▸minbnk       48
Available 16Kb banks to allow bank breakup....... 1024      3 ▸minbnkc       3




Enter  (key value) (screen#) M(enu) U(pdate) E(xit) H(elp) ▸
```

012DG100.DRW

Figure 4–10. Memory Pool Parameters

Direction Keywords

ewperc — If EWPBUF is not specified, this is the %age of the buffer pool that must be in use for the system to enter soft throttle. If EWPBUF is specified, this is the %age of buffers from the bank breakup that modifies the EWPBUF value.

ewpbuf — If not specified, this signifies that the soft throttle threshold is maintained as a %age of the total buffer pool. If specified, this value is the number of buffers that must be in use for the system to enter soft throttle, regardless of how initially specified, subsequent display is the active number of buffers that must be used for the system to enter soft throttle.

maxewp	Highest allowable value for the dynamically changing EWPBUF
arnbuf	When the system is in soft throttle, this value plus EWPBUF must again be available for the system to exit soft throttle. This is to prevent bouncing in and out of throttle.
maxbufc	If number of currently available buffers is beyond this value, do not breakup a bank into buffers.
minbufs	The number of buffers that is guaranteed to be in the system. The system tends to break banks into buffers to satisfy this number. If this value is specified in DCPOS' absolute, DCPOS comes up with the number of banks that have been broken into buffers to satisfy this value.
maxsegs	The maximum number of 4K byte segments that should be allowed in the system.
minbnk	The number of banks that is guaranteed to be in the system for program purposes. When the OS needs banks for other purposes (i.e., buffers, CACHE), banks are provided until this minimum is reached.
minbnkc	If the number of currently available banks is beyond this value, allow bank breakup.

Memory Timer Parameters

Memory timer parameters are accessed by selecting Direction Keyword 2 or mt from the @TUNER main menu. Figure 4–11 approximates the resulting screen display.

```
TUNER  5R3.01  (RSYS-7R2.00B) - Thu 15 Oct 1992 18:17:33

                        (2) Memory Timers

Memory Timer Parameters (in millisecs)       Upper-range Current Key        New
-------------------------------------------------------------------------------

Duration before breaking a bank into 128b buffers. 65535       0 ▸etbnkb       0
Duration before breaking a bank into 4Kb segments. 65535    2000 ▸etbnks    2000
Duration before releasing DCP/OS transients..... 65535    5000 ▸etosts    5000
Duration before releasing program transients..... 65535    6000 ▸etprts    6000
Duration before returning from soft throttle..... 65535     500 ▸etretn     500
Duration before terminating runs (hard throttle).. 65535  60000 ▸etrunh   60000
PP throttle pause value.......................... 65535    1000 ▸ppthrs    1000









Enter  (key value) (screen#) M(enu) U(pdate) E(xit) H(elp) ▸
```

013DG100.DRW

Figure 4-11. Memory Timer Parameters

Direction Keywords

`etbnkb` — Necessary elapsed time between bank breakups into 128 byte buffers. If value is zero, bank breakup is immediate without going through soft throttle scenario.

`etbnks` — Necessary elapsed time between bank breakups into 4K byte segments.

`etosts` — For each DCP/OS transient segment, the necessary elapsed time between its use count going to zero and releasing its space.

etprts	For each program transient segment, the necessary elapsed time between its use count going to zero and releasing its space.
etretn	Necessary elapsed time between entering and exiting soft throttle. This is to prevent bouncing in and out of throttle.
etrunh	Necessary elapsed time between entering hard throttle and the start of run terminations. If the value is zero, no run terminations occur.
ppthrs	When the system is in soft throttle, this value is used for PP throttle.

Cache/Disk Management Parameters

Cache/Disk management parameters are accessed by selecting Direction Keyword 3 or cd from the @TUNER main menu. Figure 4–12 approximates the resulting screen display.

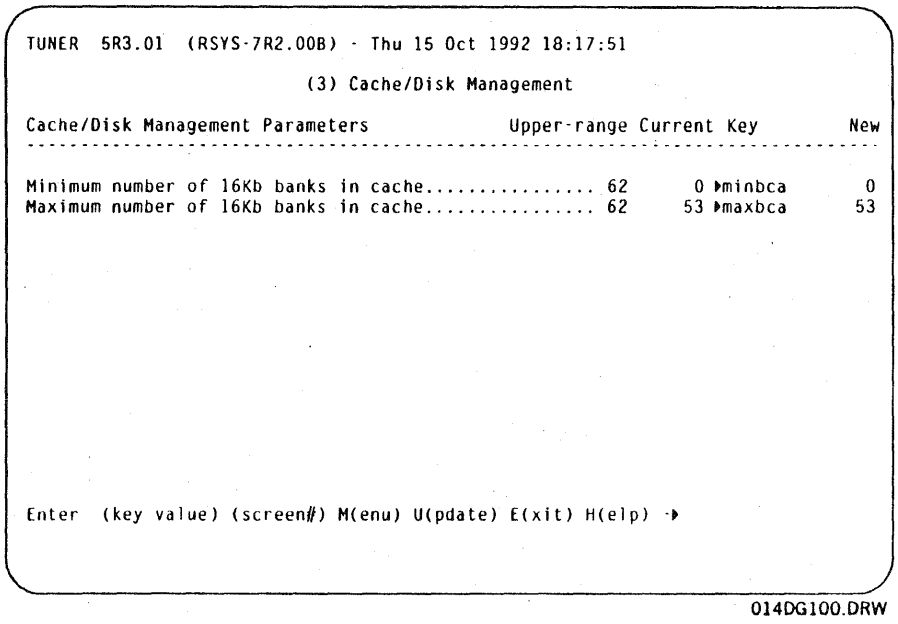


Figure 4–12. Cache/Disk Management Parameters

Direction Keywords

minbca — The number of banks that are guaranteed to be used as CACHE for mass storage. When the OS needs banks for other purposes (i.e., programs, buffers), CACHE banks are returned only until this minimum number is reached.

maxbca — The maximum number of banks that should be allowed to be used as CACHE for mass storage. CACHE management continues to receive banks until this number is reached, or there are not enough banks remaining in the system.

Run/Process Management Parameters

Run/Process management parameters are accessed by selecting Direction Keyword 4 or rp from the @TUNER main menu. Figure 4–13 approximates the resulting screen display.

```
TUNER  5R3.01  (RSYS-7R2.00B) - Thu 15 Oct 1992 18:18:10

                        (4) Run/Process Management

Run/Process Management Parameters          Upper-range Current Key      New
-------------------------------------------------------------------------------

Preemptive dispatching.....................(0-Off,1-On)      0 >prempt      0
Quantum/PN3-tick variablity................(0-Off,1-On)      0 >qtvar       0
Real time dispatching......................(0-Off,1-On)      0 >rt          0
Minimum PN3-tick time  (in millisecs)..... 65535             5 >dtmin       5
Maximum PN3-tick time  (in millisecs)..... 65535           100 >dtmax     100
Dispatch0 quantum time (in millisecs)..... 65535           100 >dtqnt0    100
Dispatch1 quantum time (in millisecs)..... 65535           100 >dtqnt1    100
Dispatch2 quantum time (in millisecs)..... 65535           100 >dtqnt2    100
Dispatch3 quantum time (in millisecs)..... 65535           100 >dtqnt3    100




Enter  (key value) (screen#) M(enu) U(pdate) E(xit) H(elp) ->
```

015DG100.DRW

Figure 4–13. Run/Process Management Parameters

Direction Keywords

prempt	Preemptive dispatching allows a higher priority task to be given CPU control even though the current task has not consumed its dispatch quantums and has not voluntarily released control.
qtvar	Need not be specified. Automatically express if variablity exists between the carious dispatch quantums and the MAX PN3-TICK (i.e., dtmax). If expressed, it may be changed as a result of actual values.
rt	Real time dispatching maintains the task's dispatch priority, in contrast to normal dispatching which places an interrupted task on the lowest dispatch queue.
dtmin	Lowest allowable value for the monitor clock which allows the OS to get CPU control periodically when the clock decrements to zero.
dtmax	Highest allowable value for the monitor clock which allows the OS to get CPU control periodically when the clock decrements to zero.
dtqnt0 dtqnt1 dtqnt2 dtqnt3	Dispatch time quantum for all tasks of that dispatch priority. A task maintains CPU control until its quantum has expired (Time-sliced), has been preempted, or has volunatrily given up control.

Log/Stats Management Parameters

Log/Stats management parameters are accessed by selecting Direction Keyword 5 or `ls` from the @TUNER main menu. Figure 4–14 approximates the resulting screen display.

```
TUNER  5R3.01  (RSYS-7R2.00B) - Thu 15 Oct 1992 18:18:38

                        (5) Log/Stats Management

Log/Stats Management Parameters              Upper-range Current Key        New
-------------------------------------------------------------------------------

Log file to be active.....................(0=No,1=Yes)      1 ▸logfil        1
Logging of file manager SVCs..............(0=No,1=Yes)      1 ▸lfsvc         1
System log file volume id.................(6 characters)       ▸logvid
Number of blocks for system log file..... 32767    500 ▸logs          500

Nominal current sample period (in millisecs)..... 999999      0 ▸period        0
Total number of sample periods needed..... 32767             0 ▸totalp        0
Start stats collection (incl low overhead)..(0-Off,1-On)      0 ▸startc        0
PN timeslice stats collection.............(0-Off,1-On)      0 ▸slice         0
Transient segments stats collection.......(0-Off,1-On)      0 ▸trans         0
Stack allocation failures stats collection..(0-Off,1-On)     0 ▸stackf        0
Disk file i/o stats collection............(0-Off,1-On)      0 ▸disk          0
PP paired messages stats collection.......(0-Off,1-On)      0 ▸ppmsg         0
Task (IPN) stats collection...............(0-Off,1-On)      0 ▸task          0
Accumulate SVC counts per run.............(0-Off,1-On)      0 ▸svccnt        0

Enter  (key value) (screen#) M(enu) U(pdate) E(xit) H(elp) ▸
```

016DG100.DRW

Figure 4–14. Log/Stats Management Parameters

Direction Keywords

`logfil`	**SYS$*SYSLOG** is enabled to receive log entries.
`lfsvc`	Selection of file manager SVCs (excludes reads and writes) to be logged.

`logvid`	When log file is made active, it is to reside on this volume. If current log file does not reside on this volume, the current file is deleted, and a new log file is cataloged.
`logs`	When log file is made active, it contains this number of blocks. If current log file is not this size, the current file is deleted and a new log file is cataloged.
`period`	The nominal cyclic time to collect and log the sample.
`totalp`	If specified, the total number of sampling cycles to be performed. When this number is reached, statistics collection is automatically turned off. If not specified (zero value), collection continues indefinitely until explicitly turned off with the parameter `startc`.
`startc`	Enable/disable the collection and logging of the sample profile as specified by the specific collection categories (automatically includes low overhead statistics).

`slice`
`trans`
`stackf`
`disk`
`ppmsg`
`task`
	Specific high-overhead collection category.

Tuning Examples

Case 1: Time Share Development

Requires a large number of banks available for concurrent testing if up to five versions of Telcon. Demand is for a debug environment with a relatively small buffer (no one is volume testing).

Tuning: set `MINBNK` = 100

Effect: Ensures that banks are available for program execution and the system does not break banks into buffers below this threshold.

Case 2: Transient Thrashing

Single program environment. System behaviour statistics indicate excessive transient loading (Telcon probably over-configured for memory size).

Tuning: set MINBUFS = xxxx
set ETPRTS = 30000 (30 secs, default is 6 secs.)

Effect: Guaranteed minimum buffer pool size (for messages and transient) and program transients are guaranteed to stay in memory for at least 30 seconds after the use-count goes to zero.

Case 3: Continuous Throttling

Statistics indicate frequent memory throttling (again, probably over-configured).

Tuning: set EWPERC = 90 (default is 80%)

Effect: The system does not go into soft throttle until 90% of the buffer pool has been allocated (for messages, tables, or transients).

Caution

There is a smaller 'safety net' backup buffer pool (10%) whenever the system goes into soft throttle. This could result in the system going into hard throttle.

4.2.46. @UBKEY — Display/Modify User Boot Keys

The UBKEY utility program displays the current User Boot Key (UBK) values and allows it to be modified. The UBK can be tested with the @IF utility. Any changes to UBK are lost when the DCP reboots, and are not seen when UBK is modified or saved to the DCP configuration file when using @MONFIG.

Format

```
@UBKEY,[options] [value]
```

Parameters

options

A	performs logical AND operation using the current UBK and value. Replaces UBK with new result.
H	prints help information.
L	prints the banner line.
O	performs logical OR operation using the current UBK and value. Replaces UBK with new result.
S	sets the UBK to value.
X	performs logical XOR operation using the current UBK and value. Replaces UBK with new result.

value is the hexidecimal value (maximum value is 0F).

Example

@UBKEY,S 0B	set user boot keys 4,3,1 and clears 2.

Section 5
Demand Mode Bypass Commands

This section does the following:

- Describes the demand mode bypass function
- Describes demand mode bypass commands
- Explains how to enter demand mode bypass commands

5.1. Using the Demand Mode Bypass Function

Demand mode bypass enables you to use designated system facilities while a program is active. You must be in demand mode to use these commands. (if you are in console mode, enter an @@END command to switch from console to demand mode.) When you use a demand mode bypass command, the processing you request runs independently of any current program execution.

For example, Table 5–1 shows how to obtain the name of a device while using the @DISK utility.

Table 5–1. Using the Demand Mode Bypass

Step	Action	Result
1	Run the utility program @DISK.	You have entered the @DISK utility and a menu is displayed.

continued

Table 5–1. Using the Demand Mode Bypass (cont.)

Step	Action	Result
2	Select option 2, Inspect Volume.	You cannot proceed until you enter the device name.
3	Enter @@CONS FS,MS.	You have entered the @@CONS demand mode bypass command, asking for the facility status of all mass storage. The names of all configured mass storage devices are listed.
4	Continue with the @DISK utility.	You are able to proceed because the @@CONS command was handled separately from the @DISK utility program; the @DISK utility program was never terminated.

5.2. Demand Mode Bypass Commands

Table 5–2 briefly describes demand mode bypass commands.

Table 5–2. Demand Mode Bypass Commands

Command	Description
@@ATT	Attach workstation to a run
@@CONS	Enter console mode while in run mode
@@CONT	Continue (release the output suspension)
@@DEB	Set the workstation in debug mode
@@DET	Detach workstation from a run
@@END	End console mode
@@FRZ	Freeze a portion of the screen
@@INS	Set the line to display scrolled output
@@NOPR	Stop printing
@@PRIV	Establish or terminate privileged mode
@@PRNT	Start printing
@@TERM	Terminate a run
@@X	Terminate a program or output

1. @@ATT — Attach Workstation to Run

The @@ATT command attaches a workstation to a specific unattached run. The specified run must be active and currently unattached. The @@ATT command detaches the workstation from the current run and attaches it to the specified run, if the specified run is active and unattached. If the specified run is not available for attachment, the workstation remains detached from any run.

By using the @@ATT command, you can attach a workstation to a batch program and then debug the program using the @@DEB command.

Format

@@ATT *run-name*

Parameter

run-name	is the name of any active run not attached to a workstation.

Example

```
@@ATT DEVNET
@@CONS RC
```

Response

```
▸ DEVNET   Run:03/Z  WS: VWS-11   Project: DEVNET Start: 00:00:59 Status: ACTV
  Program: TELCON       Tasks: 2/25 SVCs: -----   Size:    224Kb  Ports:    2
```

042DG100.DRW

Figure 5–1. @@ATT Command Output

5.2.2. @@CONS — Console Mode

The @@CONS command enables you to enter console mode commands while in demand mode. If you do not enter a command parameter, the workstation is put in console mode. All commands that are not prefixed by @ or @@ thereafter are processed as console commands, until an @@END command terminates console mode.

See Section 3 for the console mode commands you can use after entering the @@CONS command.

If you want to run a program that solicits input, be sure you are not in @@CONS mode. (You exit @@CONS mode by entering an @@END command.) Otherwise, the input is interpreted as console commands.

Format

```
@@CONS [command]
```

Parameter

command is any legal console mode command (see Section 3).

Example

```
@@CONS T
```

Response

```
@@CONS T
  Active run: BACKUP/01          PAUSE            16k
  Active run: MARY   /VWS-06
  Active run: DEVNET/03          TELCON          224k
  Active run: VACUUM/04          PAUSE            16k
  Active run: PAT    /VWS-07     DSK             576k
  Active run: RCOOK  /VWS-11
```

043DG100.DRW

Figure 5–2. @@CONS Command Output

5.2.3. @@CONT — Continue

The @@CONT command releases the output suspension invoked by the @@X O command. There can be no further output.

Format

```
@@CONT
```

.4. @@DEB — Debug

The @@DEB command sets your workstation to debug mode. See the *DCP/OS Programming Reference Manual* (7431 6894) for details of debug commands. You must be privileged to use @@DEB when a program is not running. If you are not privileged, the following message appears:

```
key-in is not allowed
```

Format

```
@@DEB
```

5.2.5. @@DET — Detach Workstation from Run

The @@DET command detaches the workstation from the current run and puts the workstation into the console mode or into no run active mode (for a virtual workstation). The user can start another run (@RUN) or attach to an active run (see the ATT console command).

Format

@@DET

Example

@RUN MARY

@@DET

T

Response

```
Active run: BACKUP/01        PAUSE          16k
Active run: MARY  /VWS-06
Active run: DEVNET/03        TELCON        224k
Active run: VACUUM/04        PAUSE          16k
Active run: PAT   /VWS-07    DSK           576k
Active run: RCOOK /*Det*
```

044DG100.DRW

Figure 5–3. @@DET Command Ouput

5. @@END — End Console Mode

The @@END command terminates demand mode bypass (activated by the @@CONS command).

Format

```
@@END
```

5.2.7. @@FRZ — Freeze Portion of Screen

The @@FRZ command holds part of the screen by setting the line at which scrolled output is deleted. The initial value is line 1 in demand mode.

If a line number greater than 23 is specified, no action is taken.

Format

```
@@FRZ line-number
```

Parameter

line-number	is the line number from which scrolled output is to be deleted. If no line number is specified, the default is 1.

Example

```
@@FRZ 3
```

8. @@INS — Set Screen Insert Point

The @@INS command sets the line at which scrolled output is displayed. The initial value is line 23.

If a line number greater than 23 is specified, no action is taken.

Format

```
@@INS line-number
```

Parameter

line-number	is the line number from which scrolled output is displayed (default is 23).

Example

```
@@INS 15
```

5.2.9. @@NOPR — Printer Off

The @@NOPR command cancels printing of console traffic on an auxiliary printer attached to the workstation.

Format

```
@@NOPR
```

10. @@PRIV — Establish Privileged Mode

The @@PRIV command establishs or ends privileged mode while running a demand mode program. See the @PRIV command in Section 4 for a description of privileged mode. @@PRIV without a password or with an invalid password ends privileged mode. READY is displayed only when the password entered is valid.

Format

```
@@PRIV [password[,newpassword]]
```

Parameters

password	is the current password; eight-character maximum. The default password when the system is booted is MONITOR.
newpassword	is entered to change the password from MONITOR (or the current password) to a new password (eight-character maximum).

Example

```
@MONFIG
@@PRIV MONITOR
```
MONFIG requires privileged mode for some functions.

5.2.11. @@PRNT — Printer On

The @@PRNT command prints a copy of all console traffic (excluding the periodic status display) on an auxiliary printer attached to the workstation.

The @@PRNT command is intended for scroll mode displays. Displayed messages are printed from the last start-of-entry (SOE) character (▸). This causes abbreviated prints when working in full-screen (menu select) mode.

Format

@@PRNT [*number*]

Parameter

number is the designated auxiliary printer number. The default is 1 (equivalent to DID 73). The range is 1 through 12.

l2. @@TERM — Terminate Run

The @@TERM command terminates the demand run and returns the workstation to console mode unless it is a virtual workstation, in which case the TELCON session is closed. It has the same effect as the @FIN command.

Format

@@TERM run-name (default is your run)

5.2.13. @@X — Kill Program or Output

The @@X command either terminates all activity of the current program or kills all workstation output until the next input. If the E option is used, a dump is iniitiated if the correct dump parameters are set.

Format

```
@@X [option]
```

Parameters

options

C	notifies program of contingency.
E	kills program with error.
I	kills input.
O	kills output.
T	kills program.
no option	defaults to TIO (kills program, output, and input).

Example

```
@@X TIO
```

Response

```
*Execution terminated*
```

Section 6
Memory Management

This section describes the following:

- DCP memory
- Disk cache
- Thresholds and throttling
- Resident and transient segments
- System tuning
- Memory management commands

6.1. Introduction

Memory is an important resource in any computer system; especially when the computer is a communications processor. In a networking environment, memory is generally used in a relatively static manner. But for message traffic, memory is used in a fragmented and volatile manner.

In this type of environment, memory must be available immediately in order to prevent messages from backing up throughout the network. It is normal to see large fluctuations in message flow from minute to minute, or even second to second. The apportioning of memory for various purposes (for example, the setting up of buffer pools and the detection of threshold conditions) is called memory management.

. DCP Memory

The DCP/OS must assign memory for several purposes:

- Global CPA tables (for example, SCT and ICT)
- Bootstrap kernel
- DCP/OS resident segments and tables
- Program code/data and CPA structures
- Program structures expansion area
- Transient segments
- Dynamic segments
- Message buffers

To meet these needs, the DCP/OS initializes memory into the following pools at boot time:

- 128K byte buffer pool

  A pool of 128K byte buffers used for program transients, dynamic segments, dynamic work areas, and message buffers.

- 4K-byte buffer pool

  A pool of 4K byte buffers used for hardware instrumentation buffers.

- 16K byte bank pool

  A pool of 16K byte buffers used for program resident segments and CPA structures.

Note: *The 4K byte buffer pool is very small and is therefore rarely used. The term buffer pool refers to the 128K byte buffer pool.*

The initial buffer pool is small, usually several hundred buffers. You can tune the system to specify a larger pool. System tuning is discussed later in this section. When the buffer pool is established, all remaining memory is placed in the bank pool.

When programs are executed, there is typically a larger demand on the buffer pool. The initial (small) pool can be quickly drained of all available memory. When this happens, the DCP/OS retrieves a spare bank from the bank pool, breaks it into 128 buffers, and places them in the buffer pool. (The demand-driven thresholds are discussed later in this section.)

The bank pool is used primarily for resident program code and CPA structures. The bank pool also acts as a write-through mass-storage cache.

Note: *It is important to recognize that many CPA structures must be contiguous in memory and often exceed 4K bytes. The DCP/OS concept of 16K byte banks does support contiguous structures larger than 16K bytes in length.*

Internal memory management routines can combine up to eight sequential banks. This provides up to 128K bytes of contiguous memory and is, for example, sufficient memory to provide a system segment table of 8191 entries. Requests for contiguous memory are transparent to an executing program and typically occur as a consequence of a program-invoked service call (SVC) to expand a particular CPA structure (for example, CPA$XLA, expand link area).

3. Disk Cache

Every record that is fewer than 1K bytes is automatically cached in spare memory banks (if available). This cache can dramatically improve the performance of disk-intensive code by eliminating the need to access the disk on subsequent reads.

All updates to records residing in cache are also written to disk, which avoids the risk of missed updates on a system crash. The cache is comprised of one or more banks from the free bank pool. Each bank is used to hold up to sixteen 1K byte pages of disk data. The oldest page is purged when no more free banks are available.

The mass storage cache is automatically enabled and expands up to a maximum of 62 banks (if available), unless the system has been tuned to override these default actions. As the system requires more banks, the DCP/OS automatically retrieves banks from the cache manager if the bank pool is empty. This action is governed by the system tuner.

6.4. Thresholds and Throttling

The buffer pool is dynamic in nature; the demands fluctuate rapidly according to changes in the network load. Because there are a finite number of buffers available, memory management is chiefly concerned with setting the buffer pool from the bank pool until there are no more banks available. When banks are no longer available, the memory manager monitors the size of the buffer pool and alerts the system when low thresholds are crossed.

There are two threshold levels that indicate low buffer pool conditions: soft throttle and hard throttle.

6.4.1. Soft Throttle

Soft throttle is entered into when the buffer pool has been drained to 20 percent of its original capacity. A soft throttle condition is not a serious situation; rather, it indicates an undesirable trend.

Note: *The soft throttle level can be set by the system tuner, either as a percentage of the buffer pool or as an absolute value.*

6.4.2. Hard Throttle

Hard throttle occurs when there are no buffers left in the buffer pool. This happens after the system has entered soft throttle and the demand for buffers has continued until the buffer pool is exhausted.

When hard throttle is triggered, the DCP/OS must wait for sufficient buffers to be returned to the pool to exit hard throttle. When sufficient buffers are returned to the buffer pool within the default time (one minute), the system returns to soft throttle.

If the system is still in hard throttle after one minute (or the value was set), the DCP/OS suspends all program dispatches, sets the PP hard throttle value, and releases the DCP/OS hard throttle pool to the buffer pool. The operating system then begins to kill the running programs, starting with the lowest run priority and going to the highest priority.

The programs are killed because if one is incapable of responding to the throttle status change alerts, it is also incapable of recovering.

Note: If the program is executing in batch mode, the runstream controls the action taken when the program is killed. Typically, a dump is performed and the program is restarted.

Programs executing on the DCP can help or hinder memory management. Any program that is a potentially large user of the buffer pool should be registered with the DCP/OS for contingency notification.

A program that is registered with DCP/OS for contingency notification is informed about every change in the buffer pool status. To work well with the operating system, programs must curtail their demands on the buffer pool when soft throttle is entered, then gradually open for full activity when notified that the system has reached normal status.

Figure 6–1 is a buffer pool profile example, showing a recoverable dip into soft throttle and a critical hard throttle situation.

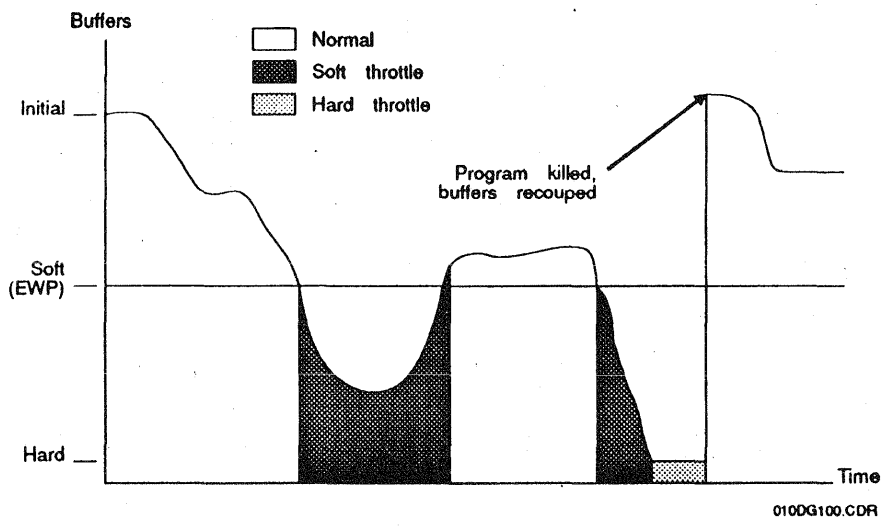


Figure 6–1. Buffer Pool Profile

6.5. Resident and Transient Segments

Resident segments are segments of code or data that reside in memory at all times. Transient segments are segments of code or data that are retrieved into memory only as required. If memory were not a critical resource, there would be no need for transient segments and all programs could be made resident. When programs are too large to be made completely resident, infrequently accessed code and read-only data segments must be marked as transient and retrieved into memory only on demand.

There are a couple of disadvantages to marking segments as transient. First, transient segments execute from subsegmented space (buffers obtained from the buffer pool). Subsegmented (transient) code does not execute as efficiently as contiguous (resident) code. The efficiency varies between DCP model.

Secondly, if a transient segment is required more frequently than projected, there is the danger of excessive swapping. Swapping occurs when the buffer pool is depleted and the space occupied by all transient segments not currently in use is returned to the buffer pool. The next request for a released segment causes that segment to be retrieved from the disk again.

There are three ways to mark a segment resident:

- Mark the segment resident in the code (SSTDEF)
- Mark the segment resident at build time
- Mark all segments resident at execution time (@PROGRAM-NAME,!)

Note: Remember that if transient segments are present, system performance may be affected. See 6.6.2 for more information on the transient segment sticking factor.

See Section 4.1.3. "Executing Programs" for additional information.

. System Tuning

Within the DCP/OS, there are algorithms that adjust the memory pools according to demands placed upon the total system. These algorithms have been designed to enable smooth adjustments to the networking load for the majority of customer applications.

There are, however, cases where the mix of load, hardware, and protocols can benefit from modifications of the system parameters within these algorithms. The modification of these parameters is called system tuning; the parameters are set by a program called @TUNER.

The @TUNER program can be run in demand or batch mode. In demand mode, a series of menus showing the current values and maximum values for sets of system parameters is displayed. In both modes (demand and batch), the system parameters may be updated either in memory or, more permanently, on the disk file containing the DCP/OS.

Note: *When a working set of parameters has been determined, it might be useful to invoke the @TUNER program from the SYSJOB.STARTUP job.*

While many parameters can be modified in the system profile, tuning the following parameters is usually most helpful:

- Minimum cache banks
- Transient sticking factor
- Initial buffer pool size
- Soft throttle level

1. Minimum Cache Banks

The minimum number of cache banks is zero, by default. This means that all banks used for the mass storage cache are retrieved from the cache manager if they are required for other purposes. If you run a disk-intensive program, ensure that a minimum number of banks are reserved for cache (to speed up file record access). To reserve cache banks, set the minimum cache banks value to nonzero.

6.6.2. Transient Sticking Factor

The transient sticking factor is the time that a transient segment is guaranteed to remain in memory after the use count becomes zero. The default value for the transient sticking factor is five seconds: The objective is to prevent excessive swapping of segments that are used regularly (but not frequently enough to be made resident).

6.6.3. Initial Buffer Pool Size

Because memory requirements are initially unknown, the initial buffer pool size is small. The system automatically increases the buffer pool on demand, adjusting to the unique network load at a particular customer site. If a larger initial buffer pool is needed, set the minimum buffer pool size to the desired value.

6.6.4. Soft Throttle Level

The soft throttle level is the system parameter most dependent on unique customer configurations and network profiles. For this reason, the @TUNER program allows you to set the throttle level to an absolute number or to a percentage of the current buffer pool. It is generally more useful to set the throttle level to a percentage rather than to an absolute number.

The soft throttle pool is set to 20 percent of the buffer pool, by default. Internal monitoring discounts any buffers used as dynamic segments in this percentage calculation (dynamic segments are usually created for the life of the program and are not candidates for reuse). As more buffers are required, the buffer pool is increased and the 20 percent figure automatically governs the size of the soft throttle pool.

. Commands for Use with Memory Management

There are several DCP/OS commands and utility programs that display information pertinent to memory management. The information focuses on memory used by programs in one of the major memory classifications (buffers and banks).

1. RC Command

The RC command displays a size field that indicates the size of the banks (in kilobytes) currently allocated to the program. The size displayed does not include the memory used for transient segments, dynamic segments, or message buffers.

2. RD Command

The RD command appends data to the RC display, detailing the amount of memory used for resident segments, transient segments, and dynamic segments. This display does not show the buffers currently allocated by the program (message buffers and work areas).

.3. T Command

The T command displays run-names, the programs that are active, and the size of the programs. As with the RC command, the size displayed by the T command represents only the banks allocated to the program.

.4. BIGB Program

The BIGB program provides a detailed breakdown of the CPA structures and banks in use by a program (for example, the number of queues and the amount of memory used for those queues).

6.7.5. SYS Program

The SYS program with the B option displays a bank map for the entire contents of memory. The run owning each bank is denoted on the map and a list of run names and numbers is also shown.

In level 4R1 and higher of the SYS program, an additional display is triggered by the M option. This display shows the amount of memory currently apportioned to buffers and banks. It shows the amount available and in use; it also breaks down information on the buffer and bank pools (for example, the number of buffers currently in use as transient segments and the number of buffers allocated in the system).

Section 7
Support

This section tells you what to do if you encounter problems with hardware or software during installation, if you have questions or comments about the documentation, or want to suggest enhancements to the documentation or software.

The DCP/OS is a fully supported software product. Therefore, it is periodically updated, revised, and enhanced. Unisys provides a software correction service for the product.

7.1. How To Report Installation Problems

If you encounter hardware or software problems during installation of the program product, contact your site administrator. When your problem requires Unisys customer support, the site administrator calls the Unisys customer support center. If a site administrator is not available and you are a customer within the continental United States, call the following toll-free number:

1-800-328-0440

Table 7–1. Unisys Customer Support Hours

Support	Hours	Days
Software	7:00 A.M. to 7:00 P.M.	Monday through Friday
Hardware	24 hours a day	7 days a week

On weekends and holidays observed by Unisys, software support is available for critical or emergency problems only.

If you are a customer outside the continental United States, you may obtain a support center telephone number from your local Unisys customer service representative. Before you call, however, read the instructions in the following manuals for more information about Unisys customer support services:

- *Support Center, Quick Reference*
- *1100/90 and 2200 Systems Support Guide*

These manuals explain services of the Unisys Customer Support Center, necessary information for the center, the priority scheme for software problems and their resolution, Remote System Support (RSS), and the User Communication Form (UCF).

If a solution to your problem is on file, the support center tells you what to do. If no record of the problem exists, the support center authorizes a UCF. The UCF process replaces the System User Report (SUR).

The support center then provides you with the following items:

- A UCF register number
- A UCF form
- Instructions for supporting materials

7.2. How to Report a Software Problem

Submit a UCF to report software problems. The *1100/90 and 2200 Systems Support Guide* describes how to submit a UCF. Use the same form to request corrections. Complete a separate UCF for each problem. The following information can help Unisys isolate your problem:

- The level of DCP/OS you are using.
- A dump or trace.
- A list of your configuration and any elements to which you have applied corrections.
- A description of what happened just before the error occurred.
- A report on the exact syntax of the last command entered before the error condition.
- A copy of the status or error messages printed at the time the error occurred (be sure to check the system console, NMS console, or DCP/OS workstation).
- A description of any files involved in the operation. You can enter `@PRT,F` *`filename`* to get a description of the files. Include the file type, also.

Check with Support Center personnel for the information they need to best diagnose and provide a solution to your problem.

3. How to Submit Corrections and Comments on Documentation

If you find an error in the documentation, or if you want to suggest improvements, you can:

- Use the business reply form at the back of each document for your comments (appropriate for brief comments).
- Submit a UCF, just as you would for a software problem or feature suggestion (appropriate for extended or detailed comments). Complete a separate UCF for each document if the problem or suggestion applies to more than one document.

These forms are designed to let Communication Systems Product Information (CSPI) respond quickly to problems and suggestions.

If you choose to submit a UCF, be sure to include the following information, which supplements the directions given on the instruction sheet attached to the UCF:

Table 7–2. UCF Information

Item No.	Item Name	What to Enter	Example
1	Class	Documentation	Documentation
2	Form	Indicate whether you are submitting a trouble report or a new suggestion	New feature suggestion
15	Product style	Product name	Telcon
16	Product level	Product level	8R1
17	Component	USER-DOC	USER-DOC
25,27	Supporting materials	Indicate if you are including document pages with the UCF	Yes, see the CD Configuration Reference Manual
35	Detailed description	Describe the documentation error or give your suggested change	

Appendix A
System Error Codes

This appendix describes system error codes which are filtered by a program that issues service requests and displays text messages explaining the nature of the error.

The error codes that follow are in the form *xxee*; where *xx* represents the error context codes assigned to each major component of DCP/OS and *ee* represents two classes: system-wide and context-specific. In general, system-wide error codes start at 0, and context-specific codes start at 80.

For example, error codes 8103 and 8403 represent two different major components of the DCP/OS. Communications Processor Architecture (CPA) services and port processor (PP) services are represented by the 81 and 84, respectively. The 03 represents a system-wide error of invalid QL.

In the error codes 8581 and 8681, the 85 and 86 represent the major components PHYSIO and LM loader, respectively. The 81 is a specific error and varies for each xx or major component.

Use the following error code tables to find the error code information you need:

Table A–11. Exec Internal Error Codes

Table A–12. Boot Error Codes

Table A–13. File Management Error Codes (returned in FR$CC)

In a full 16-bit error code, the left byte is the error context and the right byte contains the error code.

Table A–1. Error Contexts

Error Code	Description	Mnemonic
80ee	* Services - files	EC$FILE
81ee	Services - CPA	EC$CPA
82ee	* Services - run control	EC$RUN
83ee	Services - process control	EC$PC
84ee	Services - PP	EC$PP
85ee	* PHYSIO	EC$PHY
86ee	Services - line module loader	EC$LML
88ee	Loader	EC$LOAD
89ee	Services - dictionary manager	EC$DIC
8Aee	Services - inter-program messages	EC$IPM
8Bee	Services - element file package	EC$ELT
8Cee	Services - instrumentation	EC$INST
8Fee	Exec detected user errors	EC$EXEC
90ee	Services - general class	EC$SERV
EEee	†Exec internal errors	EC$EXIN
EFee	†Boot errors	EC$BOOT

* These error conditions may be accompanied by a message that suggests operation error. If an error code appears without an error message, contact a systems analyst.

† These are serious errors. Consult your Unisys systems analyst.

Refer the remaining error conditions to a programmer.

Table A-2. System-Wide Error Codes

Error Code	Description	Mnemonic
xx01	Invalid PSTx	ER$PSTX
xx02	Invalid GPLx	ER$GPLX
xx03	Invalid QLx	ER$QLX
xx04	No queue	ER$NOQ
xx05	Invalid function	ER$INVF
xx06	Invalid PN	ER$PN
xx07	Invalid SSN	ER$SSN
xx08	Invalid QT	ER$QT
xx09	Invalid QN	ER$QN
xx0A	Invalid PPID	ER$PPID
xx0B	Invalid SD type	ERSDTYP
xx0C	Invalid use count	ER$USECT
xx0D	Segment in use	ER$SSNUS
xx0E	SST full	ER$STFUL
xx0F	PT full	ER$PTFUL
xx10	PPPT full	ER$PPFUL
xx11	QT full	ER$QTFUL
xx12	SQL1 full	ER$SQFUL
xx13	Caller is not the owner	ER$OWNER
xx14	Invalid PPPT	ER$PPPT
xx15	Item is busy	ER$BUSY
xx16	Not enough memory	ER$MEFUL
xx17	Queue not emptying	ER$QDEAD
xx18	Invalid link area	ER$LA
xx19	Invalid ADR for PP start (PAO/PA)	ER$GOADR
xx1A	Queue has extant items	ER$QUSED
xx1B	Queue has invlid PN in SAI	ER$SAI
xx1C	Queu has invlid Qx	ER$QX
xx1D	No queue list	ER$NOQL
xx1E	No link area	ER$NOLA

continued

Table A-2. System-Wide Error Codes (cont.)

Error Code	Description	Mnemonic
xx1F	No PST	ER$NOPST
xx20	Queue cannot be armed	ER$QARM
xx21	Bad memory structure	ER$SEGS
xx22	Bad link area index	ER$LAX
xx23	Bad PID/PID not active	ER$PID
xx24	Process not sleeping	ER$NSLP
xx25	RUN is not privileged	ER$PRIV
xx26	No contingenccy for PID	ER$NCONT
xx27	Invalid SD specified	ER$SD
xx29	SSN not a reserved segment	ER$SSNNR
xx2A	Bad reserve SSN count	ER$RSSNN
xx2B	Invalid ICTN or PPID	ER$ICTX
xx2C	AET entry in use	ER$AETIU
xx2D	No AE list for PPID	ER$NOAEL
xx2E	Bad AEL count	ER$AELN
xx2F	AEL entry in use	ER$AELIU
xx30	No valid ALT environment table	ER$NOAET
xx31	Invalid AEL index	ER$AELX
xx32	Invalid address	ER$ADR
xx33	Timeout	ER$TIMOUT
xx34	No response from PP	ER$NRESP
xx35	Invalid lock table	ER$LT
xx36	Bad RUN number	ER$BADRUN
xx37	Service already registered	ER$SVCREG
xx38	Invalid service name	ER$VCNAM
xx39	Invalid service code	ER$ISVC
xx3A	No SDR available	ER$NOSDR
xx3B	No SST (no actrive program)	ER$NOSST
xx3C	No such service registered	ER$NSVC

continued

Table A-2. System-Wide Error Codes (cont.)

Error Code	Description	Mnemonic
xx3D	Service aborted	ER$SABRT
xx3E	A-bit not on	ER$NABIT
xx3F	Bank not found	ER$NFBNK
xx40	Illegal bank status code	ER$BNKST
xx41	Snap dump bring taken	ER$SNAP
xx42	Wrong queue type for function	ER$QTYPE
xx43	SDR present, should not be	ER$SDPR
xx44	Invalid comparison time/date	ER$CTIME
xx45	Peer side is not available	ER$PEERNA
xx46	Underlying connection failure	ER$CONNF

Table A-3. Run-Control Error Codes

Error Code	Description	Mnemonic
8280	Program terminated clearly (EXIT$)	ER$EXIT
8281	Program load error	ER$LOAD
8282	Program abort (SPEC error)	ER$SPEC
8283	Program abort (user ERR$)	ER$USER
8284	Progarm abort (operator X-OFF)	ER$XOFF
8285	Load error - no initial PN	ER$NOIP
8286	Exec-detected trap	ER$EXTRAP
8287	Bad message visibility or bad header	ER$BMSG
8288	LOG file not active	ER$LOGFIL

Table A-4. Loader Error Codes

Error Code	Description	Mnemonic
8880	SQL1 entries needed	ER$NOSQ
8881	Invalid load file (no *LD*)	ER$NDLD
8883	Program incompatible with this DCP/OS level	ER$INCP

Table A–5. PHYSIO Error Codes

Error Code	Description	Mnemonic
8580	Device not defined (invalid PDT/DVT)	ER$DNDF
8581	Device failed (PP S.I. of downed)	ER$DVFL
8582	Unrecoverable error (I/O error)	ER$UNRC
8583	Device not supported	ER$DVNS
8584	Invalid command (command, byte CT, or sector number in error, data area not present for write or improper SDs for read)	ER$IVCM
8585		ER$PPQF
8586	PP queue full (not used)	ER$DUPL
8587	Duplicate PDT entry found (device with same address already active)	ER$UEMD
8588	Unexpected media in device (density and/or other media characteristics not as expected)	ER$DVNP
8589		ER$IVDT
858A	Device not present	ER$POCF
858B	Device type connected not as in PDT	ER$LMTYP
858C	Device POC failed	ER$DVTO
8590	Line module type not compatible with device type	ER$NOCM
8591	Disk activity timeout	ER$NORM
	Can't get get cache control MCT	
	Can't get RAM bank pointer	

continued

Table A-5. PHYSIO Error Codes (cont.)

Error Code	Description	Mnemonic
8592	Size greater than allowed maximum	ER$MXSZ
8593	Unable to get banks from bank pool	ER$NOBK
8594	Unable to get banks from bank pool because of tuned memory parameters	ER$NOBKT
85A0	DD No available device	ER$DDNDSK
85A1	DD Controller sector error	ER$DDCSE
85A2	DD Both devices down at read/write	ER$DDDOWN
85A3	DD Single write at init device	ER$DDSWRT
85A4	DD Incompatible devices	ER$DDINC
85A5	DD Error at register BADSPOT	ER$DDERBS
85A6	DD Device not initialized	ER$DDNINI
85E2	Host channel down	ER$HDWN
85E4	Host channel bloaked	ER$HBLK
85E8	Workstation missing ACK	ER$MACK
85E9	Workstation bad AUX status	ER$AUXE

Table A-6. Line Module Loader Error Codes

Error Code	Description	Mnemonic
8681	Microcode load error	ER$LDERR

Table A-7. Dictionary Manager Error Codes

Error Code	Description	Mnemonic
8980	Dictionary full	ER$DCFUL
8981	Dictionary entry not found	ER$DCENF
8982	Dictionary entry already defined	ER$DCDUP
8983	Dictionary entry number out of range	ER$DCNO
8984	Dictionary packet address invalid	ER$DCPKT

Table A–8. Inter-Program Messages (IPM) Error Codes

Error Code	Description	Mnemonic
8A80	IPM entries absent for this program	ER$IPABS
8A81	Specified IPM-ID currently unused	ER$IPUSE
8A82	IPM control table full	ER$IPFULL
8A83	Duplicate receiver name	ER$IPDUPL
8A84	Receiver not registered	ER$PUNRG
8A85	No free connection available	ER$IPCONN
8A86	Invalid IPM-ID	ER$IPID
8A87	Invalid name format	ER$IPNAME
8A88	Invalid packet address	ER$IPPKT
8A89	Invalid parameter (unspecified)	ER$IPPARM
8A8A	IPM SDR present (should not be)	ER$IPSDPR
8A8B	Specified QL index already in use	ERIPQLX
8A8C	Invalid run-ID	ER$IPRUN

Table A–9. Instrumentation Services Error Codes

Error Code	Description	Mnemonic
8C81	Invalid run-ID	ER$ASID
8C82	No PPIT	ER$PPIT
8C83	IB (P) table index invalid	ER$IBX
8C84	Instrumentation not established	ER$INSNE
8C85	No IB (P) table free items	ER$NIBPS
8C86	Instrumentation internal error	ER$INTER
8C87	No procedure use table	ER$NPUT
8C88	PP already active	ER$PPA
8C89	Illegal system queue list	ER$SQLX
8C8A	Illegal access rights requested	ER$GPA
8C8B	SQL already assigned	ER$SQASG
8C8C	No SQL assigned to run	ER$NSQLA
8C8D	System soft throttle is on	ER$SOFT

Table A-10. Exec-Detected Error Codes

Error Code	Description	Mnemonic
8F80	LESG with bad mass storage address	XER$MSA
8F81	Memory not present	XER$MNP
8F82	Insufficient memory requested	XER$IMR
8F83	Bad user SDR specified	XER$BUS
8F84	Insufficient banks to meet request	XER$MRN
8F85	Return without clear wait for event	XER$WFE
8F86	Invalid memory request (get space)	XER$MRI
8F87	No memory left (get space)	XER$NOM
8F88	Bad call to PN% (services)	XER$P5E
8F89	Invalid service code	XER$IFT
8F8A	Attempt to lock same lock twice	XER$LCK
8F8B	Lock count error	XER$LCE
8F8C	SEG not released on return	XER$SNR
8F8D	Transient I/O error	XER$TIO
8F8E	Bad R1 parameter for dispatch	XER$BDR1
8F8F	Invalid flag on lock	XER$LKWI
8F90	Terminating run's task in foreign run	XER$TTFR
8F91	Terminating run has foreign task in it	XER$TRFT
8F92	Cancelled task in foreign run	XER$CTFR

Table A-11. Exec Internal Error Codes

Error Code	Description	Mnemonic
EEE0	Dispatch queue full	XER$DQF
EEE1	Fault queue full	XER$FQF
EEE2	PHYSIO queue full	XER$PQF
EEE3	I/O error (loading Exec transient)	XER$IOE
EEE4	Undefined PN4 type	XER$4ER
EEE5	Lock table error	XER$LKE
EEE6	Wait list error	XER$LST
EEE7	Call to PN2	XER$P25
EEE8	State items for same PP	XER$LNK
EEE9	PN0 trap	XER$PN0
EEEA	Wait list full	XER$WLF
EEEB	Bad segment structure	XER$SGS
EEEC	Panic stop	XER$CONT
EEED	State item for system volume	XER$SIOSV
EEEE	User requested system abort	XER$USER
EEEF	PN1 system error	XER$PN1
EEF0	Undefined PN1 type	XER$1ER
EEF1	Multi-CP stop	XER$CPS
EEF2	Professed owner not the real owner	XER$NOWN
EEF3	Bank status wrong for function	XER$BSTAT
EEF4	Not enough buffers for location	XER$NBUFS
EEF5	Bad run number being used	XER$BRUN

Table A-12. Boot Error Codes

Error Code	Description	Mnemonic
EFE0	Forced call PN0	ER$CRASH
EFE1	Forced call PN1	ER$CRASH
EFF0	Comm/line channel error	ER$COMERR
EFF1	Line module not connected (no cable)	ER$LMNC
EFF2	Illegal TOC (downloaded)	ER$ILLTOC
EFF3	Dump cannot be taken	ER$DUMP
EFF4	Unexpected response from comm line	ER$UNEXP
EFF5	Bad sector address	ER$DSL
EFF6	Dump file in use	ER$DFIU
EFF7	Dump file too small	ER$DFTS
EFF8	Dump file invalid	ER$IVDF
EFFA	Bad load path (nonexistent)	ER$BADLP
EFFB	I/O map over failed	ER$MAPO
EFFC	Line module not supported	ER$LMNS
EFFD	Download block sequence error	ER$SEQ
EFFE	Download block checksum error	ER$CHKSUM

Table A-13. File Management Error Codes (returned in FR$CC)

Error Code	Description	Mnemonic
0001	End of file	FR$EOF
0002	Word count error	FR$EWC
0003	Block number outside of file	FR$EBN
0004	Position attempt not allowed	FR$PNA
0005	Device down	FR$DWN
0006	Function not allowed	FR$FNA
0007	Disk recovery in progress	FR$RIP
0010	Catalog full	FR$CF
0011	File already cataloged	FR$FAC
0012	No space or no free DSL	FR$NS

continued

Table A-13. File Management Error Codes (returned in FR$CC) (cont.)

Error Code	Description	Mnemonic
0013	System volume read-only	FR$SRO
0016	File is not cataloged	FR$NC
0017	File in use	FR$FIU
0018	Volume is not available	FR$VNA
0019	VST not updated	FR$VNU
001A	Catalog not updated	FR$CNU
001B	Label not updated	FR$LNU
001D	Read-only file	FR$ROF
001E	Invalid FCB address	FR$IFA
001F	Open input and file empty	FR$EMP
003B	File disabled	FR$FDA
003C	Device type given instead of volume	FR$CF1
003D	Data set label not found	FR$NODSL
003E	Volume density not compatible	FR$CF3 of FR$NCD
003F	Data set label invalid format	FR$CF4
0040	Element insert - BN beyond HBW	FR$EBN1
0041	File rolled out	FR$CF6
0045	Invalid file size	FR$IFZ
0046	Invalid filename	FR$IFN
0047	Restricted file	FR$RF
0060	Invalid return queue	FR$IRQ

continued

Table A-13. File Management Error Codes (returned in FR$CC) (cont.)

Error Code	Description	Mnemonic
0062	No free FCB entry	FR$NFE
0063	Invalid MCT address	FR$IMA
0070	Nonexistent device	FR$ND
0071	Device not down	FR$DND
0072	Device already assigned	FR$DAA
0073	Device not assigned	FR$DNA
0074	Volume disabled	FR$DIS
0075	File in exclusive use	FR$FIUX
0076	Illegal temporary file access	FR$TMP
0090	Element not found	FR$EFNOTF
0091	Bad TOC block encountered	FR$EFBADT
0092	No sentinel (SDFI)	FR$EFNOS
0093	Bad record length (SDFI)	FR$EFIML
0094	Record size error (SDFO)	FR$EFSIZ
0095	Host path not configured	FR$HNC
0096	Host path not available	FR$HNA
0097	Host path is down	FR$HDWN
0098	Host path no resources	FR$HNRS

A–11. Exec Internal Error Codes (cont.)

Error Code	Description	Mnemonic
EEF3	Bank status wrong for function	XER$BSTAT
EEF4	Not enough buffers for location	XER$NBUFS
EEF5	Bad run number being used	XER$BRUN

Table A–12. Boot Error Codes

Error Code	Description	Mnemonic
EFE0	Forced call PN0	ER$CRASH
EFE1	Forced call PN1	ER$CRASH
EFF0	Comm/line channel error	ER$COMERR
EFF1	Line module not connected (no cable)	ER$LMNC
EFF2	Illegal TOC (downloaded)	ER$ILLTOC
EFF3	Dump cannot be taken	ER$DUMP
EFF4	Unexpected response from comm line	ER$UNEXP
EFF5	Bad sector address	ER$DSL
EFF6	Dump file in use	ER$DFIU
EFF7	Dump file too small	ER$DFTS
EFF8	Dump file invalid	ER$IVDF
EFFA	Bad load path (nonexistent)	ER$BADLP

continued

A–12. Boot Error Codes (cont.)

Error Code	Description	Mnemonic
EFFB	I/O map over failed	ER$MAPO
EFFC	Line module not supported	ER$LMNS
EFFD	Download block sequence error	ER$SEQ
EFFE	Download block checksum error	ER$CHKSUM

Table A–13. File Management Error Codes (returned in FR$CC)

Error Code	Description	Mnemonic
0001	End of file	FR$EOF
0002	Word count error	FR$EWC
0003	Block number outside of file	FR$EBN
0004	Position attempt not allowed	FR$PNA
0005	Device down	FR$DWN
0006	Function not allowed	FR$FNA
0007	Disk recovery in progress	FR$RIP
0010	Catalog full	FR$CF
0011	File already cataloged	FR$FAC
0012	No space or no free DSL	FR$NS
0013	System volume read-only	FR$SRO
0016	File is not cataloged	FR$NC

continued

A–13. File Management Error Codes (returned in FR$CC) (cont.)

Error Code	Description	Mnemonic
0017	File in use	FR$FIU
0018	Volume is not available	FR$VNA
0019	VST not updated	FR$VNU
001A	Catalog not updated	FR$CNU
001B	Label not updated	FR$LNU
001D	Read-only file	FR$ROF
001E	Invalid FCB address	FR$IFA
001F	Open input and file empty	FR$EMP
003B	File disabled	FR$FDA
003C	Device type given instead of volume	FR$CF1
003D	Data set label not found	FR$NODSL
003E	Volume density not compatible	FR$CF3 of FR$NCD
003F	Data set label invalid format	FR$CF4
0040	Element insert - BN beyond HBW	FR$EBN1
0041	File rolled out	FR$CF6
0045	Invalid file size	FR$IFZ
0046	Invalid filename	FR$IFN
0047	Restricted file	FR$RF
0060	Invalid return queue	FR$IRQ

continued

A–13. File Management Error Codes (returned in FR$CC) (cont.)

Error Code	Description	Mnemonic
0062	No free FCB entry	FR$NFE
0063	Invalid MCT address	FR$IMA
0070	Nonexistent device	FR$ND
0071	Device not down	FR$DND
0072	Device already assigned	FR$DAA
0073	Device not assigned	FR$DNA
0074	Volume disabled	FR$DIS
0075	File in exclusive use	FR$FIUX
0076	Illegal temporary file access	FR$TMP
0090	Element not found	FR$EFNOTF
0091	Bad TOC block encountered	FR$EFBADT
0092	No sentinel (SDFI)	FR$EFNOS
0093	Bad record length (SDFI)	FR$EFIML
0094	Record size error (SDFO)	FR$EFSIZ
0095	Host path not configured	FR$HNC
0096	Host path not available	FR$HNA
0097	Host path is down	FR$HDWN
0098	Host path no resources	FR$HNRS

Appendix B
OS 1100-Based Utilities

This appendix tells you how to:

- Convert OS 1100 absolute elements (ABSs) to DCP-formatted elements
- Build a DCP-executable ABS program
- Produce and analyze DCP dumps
- Convert OS 1100 files for use on the DCP
- Transfer dump files to or from a front-end processor (FEP) DCP

B.1. OS 1100 Processors

The OS 1100 processors are in file **DCP1100ABS** on the COMUS release tape. Table B–1 describes the functions of each processor. The processors are described in more detail alphabetically in the subsections that follow.

Table B–1. OS 1100-Based Utilities

Processor	Description
@COPYW	Copy word-addressable files
@DCPAPP	Convert OS 1100 ABS for the DCP
@DCPBUILD	Build a DCP-executable program
@DCPDUMP	Produce a formatted DCP dump
@DCPFILE	Convert OS 100 files for the DCP
@DCPFOR	Prepare a dump file for DCPDUMP
@DCPFT	FEP channel file handler utility
@DPRINT	Print in DCP/OS format

1. @COPYW

This processor translates DCP/OS files that have been transferred to a word-addressable file on the OS 1100 host with the Telcon XFER command. You can use this processor on print files, program files, and omnibus files.

Format

@COPYW[,*options*] *spec1*,*spec2*

Parameters

options

A	element type option when copying a DCP/OS program file to a Series 1100 or Series 2200 program file. If you use this option, this command copies only absolute elements.
C	copies a DCP/OS print file (for example, a print file created by LIMODUMP) to a Series 1100 or Series 2200 print file. The resulting print file can be sent to a printer.
H	displays help information.
L	lists each element name during the process of copying a program file (option P).
O	copies omnibus elements (see also option A).
P	copies a DCP/OS program file to an OS 1100/2200 program file.
R	copies relocatable elements (see also option A).
S	copies symbolic elements (see also option A).
W	skips first word of word-addressable input file. Use this option to convert old DLLCSU word-addressable files.

spec1

Option C: XFERred word-addressable file.
Option P: XFERred word-addressable file.
Not C,P : XFERred word-addressable file, to be copied to the omnibus element specified in spec2, or
: A omnibus element, to be copied to the word-addressable file in spec2.

spec2

Option C: A sector-addressable file.
Option P: A 1100/2200 program file.
Not C,P : An omnibus element, under which the contents of the word-addressable file in spec1 are to be saved, or
: A word-addressable file into which the contents of the omnibus element in spec1 are copied.

Note: *When copying elements from a DCP/OS program file to a Series 1100 or Series 2200 program file, @COPYW rounds the resulting element size to the next Series 1100 or Series 2200 sector boundary. When copying symbolic elements and print files, @COPYW attempts to correctly format each individual image into a Series 1100 or Series 2200 print line. The maximum line width is 132 characters; longer images are truncated. If an image contains carriage return characters (x'0D'), it is broken into separate print lines. All other characters in the range x'00'-x'1F' are translated to spaces.*

Examples

```
@COPYW a1stat03.,statistics.save1
```

This saves a word-addressable file into an omnibus element.

```
@ASG,T wa-stats.,d///3000000
@COPYW statistics.save1,wa-stats.
```

This restores a word-addressable file from an omnibus element.

```
@CAT,P print.,F///9999
@COPYW,C limoprint.,print.
@SYM print.
```

This converts a DCP print file to a Series 1100 or Series 2200 print file.

```
@COPYW,PSL dcplibrary.,savesym.
```

This copies all symbolic elements from a DCP program file into a Series 1100 or Series 2200 program file.

!. @DCPAPP — Convert Absolute Elements

This processor converts 36-bit OS 1100 absolute elements into DCP-formatted 32-bit OS 1100 omnibus elements.

Format

@DCPAPP,[*options*] *input-absolute*[,*output-omnibus*]

Parameters

options

	The default produces a DCP/OS omnibus element (untyped) as output.
B	builds bootstrap element.
C	builds a DCP/OS module library file (MLF) data segment containing all the names and values from the absolute value name table. User program entry points may be placed in this table using a TYPE EXTDIAG MAP directive.
D	prints debug information during processing. This does not include I/O traces.
H	displays help information.
I	prints I/O trace of input operations.
M	produces output as DCP/OS MLF (you must specify if you use the C option).
N	produces the smallest listing possible.
O	prints I/O trace of output operations.

input-absolute is the name of the absolute element.

output-absolute is the name of the omnibus element.

Note: *If the output omnibus field is not used, an omnibus element is created with the same* qual*file.element *name as the input absolute.*

Example

@DCPAPP,M TPF$.EXAM,TPF$.EXAMMLF

B.1.3. @DCPBUILD — Build ABS Program

This processor builds an absolute element (ABS) program that is executable under the DCP/OS.

Format

@DCPBUILD[,*options*] *ofile*[.*elt*],[*osdicfile.elt*],[*odicfile.elt*]

Parameters

options

B	executes in batch mode.
D	forces dictionary segments resident.
H	displays help page.
I	builds the ABS program into a file.
L	lists each processed module.
M	builds DCP/OS.
P	includes DCP/OS procedures.
R	sets read-only segments to transient and subsegmented.
S	builds a system utility program.
T	traces module details.
W	traces details of the absolute.
X	produces a cross reference listing that shows definitions and references of all CPA entities encountered during the build.

ofile.[*elt*] is the output file (or element) that contains the built ABS program.

osdicfile.elt is the input DCP/OS dictionary element (for DCP/OS segment references).

dicfile.elt — is the output element that contains a copy of the program dictionary. This element can be referenced by @DCPDUMP to process a dump of this ABS program. Note that a system dump (program load) does not include the dictionaries of programs dumped unless they have been forced resident by the D option.

Note: *Refer to the @BUILD command in Section 4 for details of the internal commands.*

4. @DCPDUMP — Analyze DCP Dumps

This processor provides a formatted DCP memory dump. It analyzes CPA and other system structures and prints the results.

Format

@DCPDUMP,[*options*] *dumpfile*.[,*dictionary element*] [,*run-name*],[*userdictionary-ID-1*, *userdictionary-element-1*]
[,... ,*user-dictionary-IDn*, *user-dictionary-elementn*]

Parameters

Note: *No options are necessary with @DCPDUMP.*

options

D	double spaces the raw dump.
H	displays help information.
M	produces a minidump: the user selects the kind of dump analysis (see Table B-2).
P	inhibits process segment descriptor register (SDR) analysis.
R	inhibits raw dump.
S	inhibits system SDR analysis.

dumpfile — is the file name of a suitably formatted DCP dump (must be formatted using DCPFOR). Normally, this file is named **NEW$DUMP**.

dictionary element

is an element containing the DCP/OS dictionary. A DCP/OS dictionary is available in the **DCP1100ABS** file on the release tape (DCPOS-DIC). DCPDUMP tries to load the DCP/OS dictionary from the file from which DCPDUMP is called (normally **DCP1100ABS**).

run-name

is an optional parameter that specifies a run name. It is useful only for a full DCP dump. This parameter permits analysis of the specified run name only. The default is to analyze all runs.

userdictionary-ID-n
userdictionary-element-n

are optional parameters that specify the user dictionary-ID and element name. On a program dump, the dump file always contains the dictionary of the dumped program to allow the proper resolution of CPA entity references. On a machine dump, the dictionaries of the running user programs may not be captured in the dump file. In such a case, provide DCPDUMP with the dictionaries. DCPDUMP accepts any number of external dictionaries on the call line. Each dictionary is represented by a pair of parameters. The first parameter, userdictionary-ID, specifies the run-ID, run name or program name for which the dictionary is to be used. The format for dictionary-ID is as follows:

<run-number>, <run-name>
or
<program-name>/P.

The second parameter, userdictionary-element, specifies the dictionary element name. Dictionary elements are created by DCPBUILD or can be extracted from the absolute by using the EXTDIC utility.

Example

```
@DCPFOR   QUAL*DUMP.

@DCP1100ABS.DCPDUMP     NEW$DUMP.
TSUM
CPST
COMT
SDR
SST
SYST
@EOF
```

Note: *It is possible to register a file using COMUS that allows the user to omit the file name if it is six characters or less. The example shows @DCPFOR without a file name, but puts in the file name for @DCPDUMP.*

The following example demonstrates how to use program dictionaries saved with @DCPBUILD:

```
@DCPFOR    QUAL*DUMP.
@DCP1100ABS.DCPDUMP,M  NEW$DUMP.,DCP1100ABS.DCPOS-DIC,,;
                  TELCON/P, SYS$LIB$*TELCON.TELCON-DIC,;
                  DSA/P, SYS$LIB$*TELCON.DSA-DIC,;
                  BLDDIB/P, SYS$LIB$*TELCON.BLDDIB-DIC
```

Dump Options

Table B–2 lists the dump selections provided by @DCPDUMP.

Table B–2. @DCPDUMP Selections

Dump Selection	Analysis
ALL	Complete dump
COMT	Common tables (CP status, process control stack, system control table, and system information)
CPQ	CP procedure queues
CPST	CP status (includes registers)
DNS	Dynamic Network Services tables
DQCN	Queue contents
DSPQ	System queue list 1
DYNQ	Queues
EHT	Error history tables
HELP	Displays possible selections
ICT	Interface control table
INSQ	Instrumentation queue
INST	Instrumentation trace
IPM	Inter-partition management tables
LACN	Link area contents
MCON	NMS configuration (MCONFIG) table (Telcon only)
MEM	Memory usage analysis
MMCB	Memory Management Control Blocks
NETC	Network connection table (Telcon only)

continued

Table B–2. @DCPDUMP Selections (cont.)

Dump Selection	Analysis
OSI	OSI program product tables
PCDT	Procedure table
PCSK	Process control stack (includes process mode active segment table)
PHEQ	PP hardware error queues
PPQ	PP program queues
PPST	PP status
PSEQ	PP software error queues
PUT	Procedure use table
RAW	Raw dump (all nonzeroed memory)
RCTN	Remote concentrator TN session table (Telcon only)
RSUM	Run summary
SCT	System control table
SDR	SDR virtual dump
SIT	System information table
SQL1	System queue list 1
SYST	Telcon debug trace (Telcon only)
SST	System segment table
TBZ	To-be-zeroed buffers
TCP	TCP/IP
TNSC	Transport connnection table (Telcon only)
TNSN	TN session table (Telcon only)
TS	Termination systems table (telcon only)
TSTK	Task stacks
TSUM	Task memory
WAIT	Wait tables analysis

B.1.5. @DCPFILE — Convert Files

This processor creates an OS 1100 omnibus element containing a DCP/OS program file structure. The omnibus element can be optionally downloaded to a DCP. The @DCPFILE uses statements in the input stream that allow certain types of information to be included in the output file. These statements are listed in Table B–3, and are described in more detail at the end of this subsection.

Table B–3. @DCPFILE Input Statements

Command	Explanation
END	Terminates input stream
IN	Controls @DCPFILE functions
TROF	Turns off trace messages
TRON	Turns on trace messages

Format

@DCPFILE[,*options*] *qual*filename.element*

Parameters

Note: *If you do not include an option, you must include a file and element name.*

options

D	creates an output file loadable by a DCP/OS based DCP.
F	creates a DCP-formatted output file.
H	requests help information.
L	displays informational messages.
S	lists each input statement (image).
T	turns on DCPFILE TRACE (turns on L and S options, also).
X	aborts if any error is found.

*qual*filename.element*

is the output file name that contains the OS 1100 omnibus element in DCP/OS program file structure.

Note: *The* D *and* F *options are mutually exclusive. When the* D *option is used, a file and element name must be used. The* F *option requires a file name, but not an element name.*

Example

```
@DCPFILE  TMP.TEST
 IN,A  DCPLIBABS.UED
 END
```

END

This statement terminates the input stream read by @DCPFILE.

Format

END [*nn*]

Parameter

nn — provides the same reserve space capability as the IN statement, except that the space is added at the end of the DCP TOC file and not at the end of an element. Also, the @DCPFILE,F and D options are not allowed.

@EOF

This statement terminates the input stream read by DCPFILE.

Format

@EOF

IN

This statement controls the type of file read by @DCPFILE.

Note: *A slash for file name keys is not supported. Only cycle values 0 through 63 are allowed.*

Format

IN,option [*qualifier**] *file.element* [*nn*]

Parameters

An option must be used or the IN statement is rejected.

options

A indicates absolute input file.

O indicates omnibus input file.

S indicates symbolic input file.

file.element file to be read.

nn is used to create additional space in increments of DCP-size blocks at the end of an element. It should be used only when the @DCPFILE,D option is used. The range of numbers giving the number of 256-byte blocks at the end of the element is 1 to 65,500.

TROF

This statement is used any time in the @DCPFILE input stream to turn off @DCPFILE trace messages.

Format

TROF

TRON

This statement is used any time in the @DCPFILE input stream to turn on @DCPFILE trace messages.

Format

TRON

.6. @DCPFOR — Format Dump File

This processor formats a dump file (retrieved from the DCP) for the @DCPDUMP processor. The output file is a temporary file named **NEW$DUMP**.

Format

@DCPFOR,[*options*] *dumpfile*.

Parameters

No options are necessary to use @DCPFOR.

options

H	displays help information.
dumpfile	is the file name of a DCP dump (to be formatted using DCPFOR).

Example

@DCPFOR QUAL*DUMP.

.7. @DCPFT — DCP File Transfer

The DCP File Transfer (DCPFT) program is a DCP/OS utility program that is used to:

- Copy files between an OS 1100 system and a DCP Series system
- Load a DCP Series system with the DCP Operating System (DCP/OS)
- Retrieve DCP/OS memory dumps
- Provide a virtual DCP/OS workstation

DCPFT can be configured by specifying parameters on the processor call line or by specifying configuration statements. The statements used to do so follow.

The DCPFT processor assigns a free DCP front-end processor (FEP) channel (word or block multiplexer) and provides a file or element transfer facility between the OS 1100 host and a FEP (the FEP must be locally attached). The file transfer is controlled using the @COPY utility on the DCP (FUP). See Section 4 for more information on the @COPY command.

Note: *If DCPFT is run in demand mode on the OS 1100 system and the host virtual workstation is configured on the DCP through @MONFIG for the channel, the demand mode terminal can be used as a DCP/OS workstation.*

In addition to file transfer, the DCPFT program can load and dump DCPs across channels. Refer to Section 2 for more information on loading and dumping. The DCPFT program can support multiple channels and multiple DCPs with concurrent file transfer across one or more of those channels.

DCPFT allows you to specify the configuration on the call line. If no information (other than options) is given on the processor call line, DCPFT assumes that the configuration reads the **READ$** file. DCP/OS does not support OS 1100 element versions or cycles.

Processor Call Options

This subsection contains the legal options which can be specified on the DCPFT call line. Options specified on the call line override corresponding parameters that have been specified in configuration statements.

Format

@DCPFT,[*options*] [*input-node/output-node*] [,*dcp-omn*] [,*dump-file*]

Parameters

options

D	writes general debug entries into the DCPFT log.
H	provides help information.
I	writes interrupt debug entries into the DCPFT log. This option must be used in combination with the D option.

L	specifies not to truncate the channel trace. Use with the D option.
T	uses the TTY presentation format for host virtual workstation input and output messages.
input-node	is the input node name used in the FEP NODE device statement. Refer to the *Exec System Software Installation and Configuration Guide* for more information on device statements. The input node must be for the channel that the DCP uses for output.
output-node	is the output node name used in the FEP NODE device statement. Refer to the *Exec System Software Installation and Configuration Guide* for more information on device statements. The output-node must be for the channel that the DCP uses for input. For a pseudo full-duplex path, output-node need not be specified. If output-node is specified, it must be the same as the value for input-node.
dcp-omn	is the DCP omnibus load element. This element contains the DCP/OS to be transferred to the DCP during a cross-channel load operation. If cross-channel loads are not performed, this parameter may be omitted.
dump-file	is the file name used to catalog DCP dump files. When a cross-channel dump operation is performed, this file is cataloged with a +1 cycle and the DCP memory dump is written to this file. The default file name is **[*input-node*]$DUMP$**.

Note: ***Input-node*** *is used as the single path name when you use the processor call line to configure DCPFT.* ***DCPFTLOG*** *is used as the name of the DCPFT log file.*

Configuration Statements

The configuration is made up of configuration statements. If no parameters are entered on the call line, the configuration statements are read following the call line.

PATH Configuration Statement

The PATH configuration statement is used to define a physical connection to a DCP. Multiple paths may be defined by specifying more than one path statement.

At least one PATH configuration statement must be specified (assuming that the configuration has not been specified on the call line).

When an attempt is made to initialize a path and the path cannot be brought up, DCPFT initializes the path in a down state. You can define two or more different paths that have the same input and output nodes specified; however, only one path can be up at a time.

Format

PATH,*path-name*	INPUT-NODE,*input-node* OUTPUT-NODE,*output-node* LOAD,*file.elt* (required if performing cross-channel loads) DUMP,[*dump-file*] STATUS,[*up-or-down*] TRACE,[*high-low-or-off*]

Parameters

PATH,*path-name*	names the path to the DCP. It can be from 1 to 6 alphanumeric characters long. Each name must be different from any other name in the configuration. Uppercase and lowercase letters are equivalent for the purposes of determining name uniqueness. The path name identifies the path in DCPFT log entries as well as on the UP, DOWN, and STATUS commands. The default path name is input-node.
INPUT-NODE,*input-node*	is the input-node name used in the FEP NODE device statement. Refer to the *Exec System Software Installation and Configuration Guide* for more information on the FEP NODE device statement.

OUTPUT-NODE,*output-node*	is the output-node name used in the FEP NODE device statement. Refer to the *Exec System Software Installation and Configuration Guide* for more information on the FEP NODE device statement. For a pseudo full-duplex path, output-node is specified with the same name as input-node.
LOAD,*dcp-omn*	is the load element sent to the DCP during a cross-channel load. The load element is an omnibus element containing the operating software for the DCP. It is specified as a program file element.
DUMP,[*dump-file*]	is used to specify the name of the OS 1100 file used to receive DCP memory dumps. When a cross-channel dump operation is performed, this file is cataloged with a +1 cycle and the DCP memory dump is written to this file. The default file name is **[input-node] $DUMP$**.
STATUS,[*up-or-down*]	is specified as either UP or DOWN and indicates the logical status of the path when DCPFT is initialized. The default is UP.
TRACE,[*high-low-or-off*]	is either HIGH, LOW, or OFF and is used to control logging of input and output channel messages in hexadecimal format to the DCPFT log file. HIGH logs entire messages. LOW truncates messages after 32 bytes. OFF produces no trace. The default is OFF.

INFO Configuration Statement

The INFO configuration statement specifies general information pertaining to the execution of DCPFT. All parameters in this configuration statement are optional.

Format

INFO

LOG,[*log-file*]
HVWS-TYPE,[*tty-or-uni*]
FILE-ASG,[*asgx-or-noasgx*]
RT-LEVEL,[*rt-level*]
MAX-FILES,[*max-files*]
DEBUG,[*on-or-off*]

Parameters

LOG,*log-file*	specifies the name of the DCPFT log file. This file is cataloged +1 cycle when DCPFT is initialized. The default filename is **DCPFTLOG**.
HVWS-TYPE,*tty-or-uni*	must be either TTY or UNI (UNISCOPE®). This parameter is used to control the presentation format of output text messages to the host virtual workstation. It is also used to process the input messages from the host virtual workstation. TTY is used if DCPFT is executed by a demand run which is being controlled by using ER RSI$. The default is UNI.
FILE-ASG,*asgx-or-noasgx*	is either ASGX or NOASGX. This parameter is used to indicate whether exclusive file assignment is used when DCPFT processes a DCP/OS file request packet to open a non-program file for writing. ASGX indicates that exclusive file assignment (@ASG,AX) is performed. NOASGX indicates that non-exclusive file assignment (@ASG,A) is performed. The default is ASGX.
RT-LEVEL,*rt-level*	ranges from 2 (highest priority) to 35 (lowest priority) and must be specified as a decimal number. This value determines the real-time priority assigned to DCPFT. The default is 2.
MAX-FILES,*max-files*	The range is from 0 to 35 and must be specified as a decimal number. This value specifies the maximum number of files which may be opened by the aggregate of DCPs with paths connected to DCPFT. When DCPFT is initialized, memory space is reserved for performing file transfers. DCPFT reserves approximately 1000 words for each file opened (35 files = 35K). By specifying 0, you prevent DCPFT from performing file transfers, while allowing the host virtual workstation to function. The default is 35 files.

DEBUG,*on-or-off* must be either ON or OFF. When ON, debug entries are written to the DCPFT log file during the execution of DCPFT. When you specify ON, it is the same as specifying the D option on the call line. The default is OFF.

ENDCONFIG Configuration Statement

The ENDCONFIG configuration statement indicates the end of the configuration statements.

Format

ENDCONFIG

DCPFT Commands

This subsection describes the commands that control the execution of DCPFT, whose commands entered through the **READ$** file must be preceded by two forward slashes (//). DCPFT commands entered from an OS 1100 console are not preceded by two slashes.

The UP, DOWN, HELP, STATUS, and TERM commands are available from the OS 1100 console interface. The host virtual workstation (HVWS) interface is not supported from the OS 1100 console.

UP

The UP command brings a path to a logical ready state. The channel input node and channel output nodes are assigned, then initialized using the FEPI function.

Format 1

UP *path-name*

Format 2

//UP *path-name*

Parameter

path-name is the name of the logical path to the DCP as defined in DCPFT.

DOWN

The DOWN command brings a path to a logical not-ready state. The channel input node and channel output nodes are terminated using the FEPT function. They are then freed. The channel can then be assigned and used by other runs.

The DOWN command also causes all resources associated with the path to be reset. For example, if a file transfer is taking place when a path is downed, the file is freed. All output queued to the DCP on the disabled path is purged.

Format 1

DOWN *path-name*

Format 2

//DOWN *path-name*

Parameter

path-name	is the name of the logical path to a DCP as defined in DCPFT.

HELP

The HELP command is used to solicit help. This displays a list of other DCPFT commands to be displayed.

Format 1

HELP

Format 2

//HELP

STATUS

The STATUS command queries the status of a path to a DCP. The path may be either UP or DOWN. If the path is UP, it may be in one of four states: active, loading, dumping, or inactive.

Format 1

STATUS *path-name*

Format 2

//STATUS *path-name*

Parameter

path-name is the name of the logical path to the DCP as defined in DCPFT. If the path-name is omitted, the status of each configured path is displayed.

TERM

The TERM command is used to terminate DCPFT. The //TERM command can be used after configuration processing to terminate DCPFT.

Format 1

TERM

Format 2

//TERM

CONNECT

The CONNECT command is used to make the host virtual workstation active on a given path. If the host virtual workstation was previously active on a path, it is logically switched to the new path.

To enter demand mode commands (@) and demand mode bypass commands (@@), enter / and //, respectively, to pass the OS 1100 interpretation of these commands (when passed OS 1100, the DCPFT translates the / or // into @ or @@, respectivley).

If a single path is configured in DCPFT, the //CONNECT command is not required to make the host virtual workstation active. The host virtual workstation is automatically set to the single configured path.

Note: *The host virtual workstation is not supported from the OS 1100 console.*

Format

//CONNECT path-name

Parameter

path-name is the name of the logical path to the DCP as defined in DCPFT.

END

The END command causes DCPFT to stop soliciting input from the OS 1100 console.

Format

END

Note: *The END command is supported only from the OS 1100 console.*

DATE

The DATE command is used to read the date and time on the OS 1100 and send an input message from the active host virtual workstation.

Note: *The DATE command is not supported from the OS 1100 console.*

Example of Single Channel Assignment

```
@DCP1100*ABS4R1.DCPFT F403
```

Example of Multiple Channel Assignment

```
@DCP1100*ABS4R1.DCPFT
PATH,F401  INPUT-node,F401  OUTPUT-node,F401
PATH,F402  INPUT-node,F402  OUTPUT-node,F402  status,DOWN
PATH,F403  INPUT-node,F403  OUTPUT-node,F403
PATH,F404  INPUT-node,F404  OUTPUT-node,F404
PATH,F4012 INPUT-node,F4012 OUTPUT-node,F4012 status,DOWN
PATH,F502  INPUT-node,F502  OUTPUT-node,F502  status,DOWN
PATH,F503  INPUT-node,F503  OUTPUT-node,F503
```

Example of Load

```
@DCP1100*ABS4R1.DCPFT F403,SYS*SYSTEM.LOAD
```

Example of Dump

```
@DCP1100*ABS4R1.DCPFT F403,SYS*SYSTEM.LOAD,DUMPFILE.
```

Example of Virtual Workstation Mode

Runmode for HOSTVWS

```
//CONNECT path-name
/RUN run-id
/DISK
//CONS SS
//CONS T
/PRT
/FIN
```

Additional Discussion

Using @DCPFT requires that you have a large program control table (PCT) (/K option), and that you operate in real time. You must have absolute device assignment when using @DCPFT. The run quota must be greater than 100,000 words of memory.

8. @DPRINT — Print Sector and Word-Addressable Files

This utility prints sector and word-addressable files, as well as omnibus and absolute elements, in DCP/OS (displays 256 bytes at one time) sector format. It also provides a simple string search function.

Format

@DPRINT[,options] *file*[*,elt*][*,start*][*,end*]

Parameters

options

A	prints or searches an absolute element for a character string.
B	prints 132-column lines (default is 80 characters).
F	terminates string search after first occurrence of string.
H	displays help information.
O	prints or searches an omnibus element for a string.
S	searches an element or a file for occurrences of a string. The @DPRINT utility asks for the search string in a separate query when this option is specified.
W	skips first Series 1100 or Series 2200 word in word-addressable file. This option is used in conjunction with old DLLCSU files.

file — prints or searches the contents of a sector or word-addressable file.

file.elt prints or searches the contents of an absolute (for example, a DCP program) or omnibus (for example, MLF) element. Without the A or O options, @DPRINT attempts to locate an omnibus or absolute element (in that order) to process.

start is the starting sector number (default is 0).

end is the ending sector number (default is highest sector in the file or element).

Note: *The @DPRINT utility queries for a search criterion when the* S *option is specified. The criterion can be specified in hexadecimal or as a case-sensitive character string. If the string contains embedded spaces, it must be specified in quotes.*

Examples

@DPRINT mlffile.mymlf,0,9

This prints the first 10 sectors of an MLF.

@DPRINT,BO mlffile.mymlf,0,9

This prints the first 10 sectors of an MLF in wide 132-column format.

@DPRINT,SF dumpfile.

This stops on first occurrence.

'CFACCS TRACE'

This searches a retrieved dump file for the specified string 'CFACCS TRACE'. The search terminates after the first occurance. If a string spans two sectors, it will not be found. You can, however, search on a partial string.

option is specified. The criterion can be specified in hexadecimal or as a case-sensitive character string. If the string contains embedded spaces, it must be specified in quotes.

Examples

@DPRINT mlffile.mymlf,0,9

This prints the first 10 sectors of an MLF.

@DPRINT,BO mlffile.mymlf,0,9

This prints the first 10 sectors of an MLF in wide 132-column format.

@DPRINT,SF dumpfile.

This stops on first occurrence.

'CFACCS TRACE'

This searches a retrieved dump file for the specified string 'CFACCS TRACE'. The search terminates after the first occurance. If a string spans two sectors, it will not be found. You can, however, search on a partial string.

Appendix C
Non-Supported Utilities

There several useful utilities (not critical to the operation and maintenance of DCP/OS) in the **DCPLIBABS.** file on the release tape. Use the H option to get additional information on these utilities, which are described in Tables C–1 and C–2.

Table C–1. DCPLIBABS.file Utilities

Utility	Description
@BACKUP	Manager for securing files
@BIGB	Program structure display
@BLOCK	Display blocked characters
@BOOTER	DCP boot facility
@CONS	@@CONS in batch mode
@DIR	qual*file.elt directory display
@EXTDIC	Extract program dictionary
@FCOMP	File or element compare
@FUNC	Interpret SVC function code
@GREP	Pattern utility (Unix-like)
@HDCOPY	HDLC link file transfer
@IFPREP	IFDC diskette format utility
@IPM	Display inter-program users
@IPMTST	Inter-program message test
@LD	Display loadable (ABS) element details
@LIST	List/modify contents of file/element
@MLF	Display MLF (OMN) element details
@NEWS	Display news messages
@PAUSE	JCL pause
@POSTBUG	Display details from last boot
@TYPE	Display source file/element in pages
@UED	Full screen editor
@UNDELETE	Undelete element

Table C–2. Nonsupported OS 1100 Utilities

Utility	Description
@DCPLOAD	Cross channel load/dump

Appendix D
Workstation Messages

This appendix lists and describes the error messages and advisory messages you may see at your workstation. Operator actions and recovery procedures are provided with message descriptions where appropriate.

rrrrrr(*nn*)/*ppppppp* **Abort FIN** *hh:mm:ss*

Explanation

The run name *rrrrrr*, run-sequence number *nn*, terminated in error: *ppppppp* was the active program. If no program was active (for example, if you entered TERM RUNID on an inactive run, *ppppppp* would be absent).

Add Stack Is Full - Add Statement Ignored

Explanation

You stacked too many @ADD commands. The system discarded your last input.

Operator Action

Use fewer @ADD commands.

Auxiliary Device Error

Explanation

The system detected an auxiliary device error. The device is usually the printer attached to the workstation.

Operator Action

Make sure the printer is online and configured correctly.

Auxiliary Device Not Configured

Explanation

You attempted to use an auxiliary device that is not configured correctly for your workstation.

Operator Action

Use @MONFIG to validate the configuration and try again.

Bad Run Statement

Explanation

You entered an @RUN statement with incorrect syntax.

Operator Action

Check the appropriate section of this manual, correct the statement, and reenter it.

Catalog Full - Some Files Not Registered

Explanation

The system catalog is full.

Operator Action

If you need all the files available from the current volume, investigate the possibility of taking down other volumes.

cccccccc **Channel Interface Active Time:** *hh:mm:ss*

Explanation

The channel with a PDT name of *cccccccc* is now in connect mode. This is an advisory message only; the channel is available for use.

*** Console Output Lost ***

Explanation

The system detected that console output to the workstation was discarded. Normally, this is the result of a heavy output load to the workstation, a workstation error, or the use of message waiting.

Data Ignored - In Control Mode (Use @HELP)

Explanation

An entry was made in demand mode when no program was active to receive the data.

Operator Action

Check your input. It may lack a leading @, or the program that was to receive the data may be closed out.

Disconnect Occurred - *cccccccc* Channel Interface Inactive Time:*hh:mm:ss*

Explanation

The channel with a PDT name of *cccccccc* went into disconnect mode.

Operator Action

Check the host system and verify if the peer program is disabled.

***wwwwwwww* Does Not Exist, TM Ignored**

Explanation

The indicated workstation does not exist, so the terminal message (TM) was discarded.

Operator Action

Use the T command to determine active workstation names.

DN Key-in: *wwwwwwww* is in Interactive Mode, Key-in Ignored

Explanation

You entered a DN command with the workstation named *wwwwwwww* in demand mode. The system allows workstations to be taken down only if they are in console mode.

Operator Action

If necessary, use the T command to determine the run name that is active on the workstation, then use the TERM run name to return the workstation to console mode.

DN Key-in: *dddddddd* is the System Volume, Cannot Be Downed

Explanation

You entered a DN command in error. The specified device is configured but is currently the volume containing DCP/OS transients. Bringing down the system volume (making it offline) would be catastrophic to the operating system.

Operator Action

If you need to take down this volume, you must reboot the DCP from a different volume (different LPT) or you can use @MONFIG to force the system to be entirely resident (in this case there is no reserved system volume). If you force the system volume to be totally resident, slightly more memory is required.

runid: **Dump Errored. Status=[***ssss***]** *xxxxxxxxxx*

Explanation

The dump taken for the specified run-ID was not completed correctly.

Operator Action

The error status *ssss* can be decoded using @FAC. However, *xxxxxxxxxx* normally provides the explanation. Typically, an incorrectly completed dump is caused by a lack of mass storage. Make sure all unneeded files (particularly dump files) are purged.

runid: **Dump Taken To File** *q***f*

Explanation

A dump was taken for the specified run-ID. Check the dump with @DMPI or @IDUMP to possibly discover the reason for the dump. If more long-term analysis is required, the file can be transferred to a suitable host to obtain a hard copy. Whatever you do, delete the dump file as soon as it is free, since dump files tend to occupy relatively large amounts of mass storage. Note that when using normal options (see @CRASH), the system does not overwrite an existing dump file. Consequently, subsequent dumps can be lost if there is insufficient mass storage.

runid: **Dumping...**

Explanation

The run indicated is currently attempting to dump a program.

spec 1 **- Element Name Illegal**

Explanation

The command you entered has the format *@aaaa.bbb.ccc*. The *ccc* (element name or subfield 1) is invalid.

Operator Action

Review the instructions for the command in this manual. The option you used (*@aaaa,option bbb.ccc)* may have caused the error.

spec 1 - **Element Name Required**

Explanation

The command you entered has the format *@aaaa bbb*. An element name or subfield 1 is needed (@aaaa bbb.ccc).

Operator Action

Review the instructions for the command in this manual. The option you used (*@aaaa,option bbb.)* may have caused the error.

Element Not Found. Status=[*ssss*] *xxxxxxxxxxxxxxxx*

Explanation

The system searched for the element requested and could not locate it using the normal system algorithms. The error status *ssss* can be decoded using @FAC; however, *xxxxxxxxxxxxxxxx* normally provides the explanation.

Element Type(s) Not Found

Explanation

You requested specific element types (normally O option for omnibus [OMN], S option for symbolic [SYM], and A option for absolute [ABS]) but the specific types were not found.

Operator Action

Check for the proper option and *file.elt* usage.

End Ignored - In Control Mode

Explanation

You entered an @END command when none was expected.

Operator Action

Check the state of the program for which the command is intended.

Enter Date And Time - D 90-12-03 *hh:mm:ss*

Explanation

A DCP was rebooted or a D command entered with no date/time confirmed.

Operator Action

Enter the date/time as: YY-MM-DD HH:MM:SS

EOF Ignored - In Control Mode

Explanation

You entered an @EOF command when none was expected.

Operator Action

Check the state of the program for which the command is intended.

rrrrrr(*nn*)/*pppppppp* **ERROR** *hh:mm:ss*

Explanation

The program *pppppppp* has a contingency error under run name *rrrrrr* (run sequence number *nn*).

rrrrrr(*nn*)/*pppppppp* **ERROR FIN** *hh:mm:ss*

Explanation

The run name *rrrrrr* (run sequence number *nn*) terminated in error: *ppppppp* is the active program. If no program is active (for example, if you entered TERM RUNID for an inactive run), *pppppppp* is absent.

Error In Port/Line

Explanation

The port number you provided is invalid.

Operator Action

Enter the command FS,PORTS to display all valid port numbers.

Error: Resources Too Low To Increase Buffers

Explanation

Your request to increase the free buffer pool was not acted on because the available resources are too low. Always be careful about manually depleting resources. The system attempts to balance all resources. Manually overriding these allocations can cause undesirable effects. Consult your systems analyst if necessary.

runid: **Execution Aborted: E-Type=***tttt* **E-Code=***cccc* **PN=***pppp* ADDR=*aaaa*

Explanation

The indicated run name has a program contingency. The error type *tttt* and the error code *cccc* can be analyzed with @FAC. The CPA procedure number for the faulting program is *pppp*. The contingency occurred at the virtual address *aaaa* in the procedure *pppp*.

Execution Terminated

Explanation

The system terminated execution of the current program.

Facility Reject. Status=[*ssss***]** xxxxxxxxxxxx

Explanation

A facility status error was detected. You can use @FAC to decode the status (*ssss*) but the explanation *xxxxxxxxxxxx* should be clear.

File Error. STATUS=[*ssss***]** *xxxxxxxxxxxxxxxxxxxxx*

Explanation

A file error was detected. You can use @FAC to decode the status (*ssss*) but the explanation *xxxxxxxxxxxxxxxxxxxxx* should be clear.

rrrrrr(*nn*)/ FIN *hh:mm:ss*

Explanation

The run name *rrrrrr* (run-sequence number *nn*) processing is complete.

@FIN IN Add Stream - Ignored

Explanation

The add stream you executed includes an @FIN statement in it. Under normal conditions, this would terminate the run. However, since this command is in an @ADD stream, it is ignored.

Interrupt

Explanation

The system responded to your use of the attention key (break).

Operator Action

You may now proceed.

Invalid @@ Command

Explanation

A command you entered that is prefixed by @@ (or ##) is not a valid system escape mode command.

Operator Action

See Section 5 for valid @@ commands.

Invalid Format or Values - Input Ignored

Explanation

The system or program cannot analyze your entry.

Operator Action

Check that the command you entered is valid and that you followed the prescribed syntax.

Invalid Parameter - Input Ignored

Explanation

The program cannot analyze the parameters you specified.

Operator Action

Check that the command you entered is valid and that you followed the prescribed syntax.

dddddddd I/O Error=*eeee* FUNC=*ffff* ADR=*aaaa* LMS=*ssss* DVS=*ssss* SPS=*ssss*

Explanation

A mass storage input or output error was detected for PDT device name *dddddddd*. The error code *eeee* can be decoded using @FAC. The remainder of the message provides information on the function (*ffff*) performed on the mass storage and with the sector address *aaaa* at which this error was detected. The line module status (LMS), device status (DVS), and supplementary status (SPS) are also included. This information is also held in the I/O error log, which can be inspected using the @IOL utility. This utility also decodes the LMS/DVS/SPS on request, giving an engineer further information on the problem.

Operator Action

Before calling for assistance, make sure the device is physically and electrically online. Also make sure the media used are available, formatted correctly, and inserted properly.

* IO-ERR Device:*dddddddd* Status:*ssss* Time:*hh:mm:ss* - Message Discarded *

Explanation

An I/O error occurred for the PDT device name *dddddddd*. The error status can be decoded using @FAC.

Operator Action

Make sure that the device is physically and electrically online. Also make sure the media used are available, formatted correctly, and inserted properly.

dddddddd **Is Not a Workstation, TM Ignored**

Explanation

The indicated device is not a workstation, so the terminal message (TM) was discarded.

Operator Action

Use the T command to determine active workstation names.

cc **Key-in:** *dddddddd* **Already Down, Input Ignored**

Explanation

You entered a DOWN command for a device named *dddddddd* (PDT name) that is already down.

xx **Key-in Does Not Exist, Input Ignored**

Explanation

The key-in entered is not known to the DCP/OS.

Operator Action

Enter the correct key-in.

cc **Key-in:** *dddddddd* **Does Not Exist, Input Ignored**

Explanation

You entered a command for a device named *dddddddd* that is not configured in the PDT.

cc **Key-in:** *dddddddd* **Must Be Down First, Input Ignored**

Explanation

You entered a command for a device named *dddddddd* that must be placed in an offline mode before attempting the command. This requirement is normally enforced to ensure that no one is using the device *dddddddd*.

cc **Key-in Not Allowed at This Time - System Not Initialized**

Explanation

The command you entered is not appropriate at this time. This message should appear only when the system is being initialized and is unable to accept console requests.

Operator Action

Wait until the message Mass Storage Recovery Complete appears. Then reenter the command.

cc **Key-in: (*oooooooo*) Option Does Not Exist, Input Ignored**

Explanation

You entered a command for the option named *oooooooo*, which is not supported.

Load Aborted: E-Type=*tttt* E-Code=*cccc*

Explanation

The system loader cannot load a program.

Operator Action

Use @FAC for more information on the error-type (*tttt*) and the error-code (*cccc*).

Mass Storage Recovery Complete

Explanation

All recoverable mass storage was recovered according to the configuration. The system can now accept commands. (This message appears on all configured workstations after a system reboot.)

spec nn - **Max Number of Characters Exceeded**

Explanation

A field (parameter), subfield, or a number *nn* was too long for the command entered.

Operator Action

Review the command instructions on valid syntax.

Maximum Number of Batch Jobs Already Active

Explanation

The amount of concurrency of active batch mode jobs was exceeded and the system aborted the job that was started.

Operator Action

Use the MX command to increase the batch concurrency. Note that excessive concurrencies can deteriorate system throughput, depending upon the use of each run.

Maximum Number of Interactive Jobs Already Active

Explanation

The concurrency of active demand mode jobs was exceeded. You attempted to enter an @RUN command, which the system rejects.

Operator Action

Use the MX command to increase the demand concurrency. Note that excessive concurrencies can deteriorate system throughput, depending upon the use of each run.

No Run Active

Explanation

You entered data with a leading @ character when the workstation is not in run mode.

Operator Action

Examine your input and remove the @ sign, if necessary, or enter a valid @RUN command.

rrrrrr **Not Found**

Explanation

The run name *rrrrr* cannot be found.

Operator Action

Use the T command to determine currently active runs.

rrrrrr(*nn*)/*pppppppp* **Out of Tasks**

Explanation

For the given run name *rrrrr* (run sequence number *nn*), the program *pppppppp* requires more tasks to coexist than was defined for this program. Normally, this indicates an error in the program.

Operator Action

If the messages persist, a dump or other diagnostic information should be collected. The RT command gives details of all current tasks that are active. Your systems analyst can use the information to determine the cause of the error. In a development environment, this problem can be caused by using incorrect @BUILD parameters. These parameters can be corrected by entering @PROGRAM,///*nstk*, where *nstk* is the correct number of stacks required.

Occasionally, this message appears for other reasons. It can be caused by a mass storage hang (where most of the tasks are waiting for I/O), or lack of memory (where most of the tasks are waiting for allocation of buffers). In general, if you do not know what the problem is, dump the program. In any case, terminate the program by using the E command and call a systems analyst for advice.

*** Error on port pppp - Type** *nn/nn* **(info=xx) at** *aaaa* **- Time:** *hh:mm:ss* *****

Explanation

An unexpected CPA PP state item occurred for port *pp*. Further analysis defines the CPA error code (*nn/nn*) supplementary information (XX) at the virtual address *aaaa*. In general, this indicates that the stated port is no longer available to the DCP/OS.

Operator Action

Try to recover the port using the UP command. Use @IOL to display all recent DCP/OS disk PP state items. Use @FAC,P to interpret PP type CPA error codes.

Program Not Found. Status=[*ssss***]** *xxxxxxxxxxxx*

Explanation

The system searched for the program requested and cannot locate it using the normal system algorithms. The error status *ssss* can be decoded using @FAC. However, *xxxxxxxxxxxx* normally provides the explanation.

spec nn **- Required Fields or Subfields Missing**

Explanation

You did not enter a field (parameter), subfield, or number *nn* when one was required for this particular command.

Operator Action

Review the command instructions for valid syntax.

Run Already Active

Explanation

You entered an @RUN command on a workstation that is still in demand mode.

Operator Action

Either ignore this message or enter @FIN followed by a suitable @RUN command.

*** S R E CODE:***cccc* **MTYP:***mmmm* **SUBF:***ssss* **ID:***iiii* **TIME:***hh:mm:ss* *****

Explanation

A system recoverable error occurred. Interpret the error code from the following chart:

cccc (error code)

1 message type unknown
2 invalid workstation number
4 workstation disconnected
5 no user identifier available

mmmm is the message type

ssss is the subfield type

ID is the workstation identifier

In general, error codes 1 and 5 imply some form of software error. Error codes 2 and 4 normally occur just after a workstation is disconnected or disabled and there is still data queued for it in the system.

Operator Action

If the situation occurs repeatedly, check the workstation connection.

rrrrrr*(*nn*)/ Start *hh:mm:ss

Explanation

The run name *rrrrrr* (run sequence number *nn*) started.

T/D 1990-12-03 *hh:mm:ss*

Explanation

This message appears every six minutes if the time and date are set.

Terminated Due to Unavailable System Resources

Explanation

The total amount of concurrency of all active jobs was exceeded and the system aborted the job that was started.

Operator Action

Determine whether selected jobs can be aborted (using the TERM command). To avoid this situation, use the MX command to keep the concurrency at a manageable limit.

* U D M Device:*dddddddd* Status:*ssss* Time:*hh:mm:ss* -Message(s) Discarded *

Explanation

The DCP/OS was unable to output a message to the PDT device name *dddddddd*. The reason is given as status code *ssss*, which can be interpreted using @FAC.

UP Key-in: *dddddddd* **Cannot Be Enabled, Status:** *eeee*

Explanation

You entered a command for a device named *dddddddd* (PDT name) that cannot be brought into a ready condition. The reason is given as an error code *eeee* that, like all four-digit error codes, can be analyzed using the @FAC. This message is normally followed by the dddddddd DN Port:pppp message, which confirms that the device was returned to a not ready condition.

Operator Action

Check that the *pp* (port number) is the number expected. If it is not, you can use the MV command to move the device.

**** Undefined Canned Message ****

Explanation

You attempted to send a canned message, but specified an invalid message number. (This message implies a software error in the program being executed.)

dddddddd Up Port:*pppp* LM:*llll* MC:*mmmm* CTL:*c* DRV:*d* TYP:*tttt* **VOL:***vvvv*

Explanation

You attempted to use a DN command on mass storage that is in use. The DCP/OS is preventing the device from being taken offline, since there are runs currently using that mass storage.

Operator Action

If you must complete this command, you should terminate suspected runs. This message normally appears as an advisory message after an UP command of a mass storage device.

runid: **User Aborted: E-Type=***tttt* **E-Code=***cccc* **PN=***pppp* **ADDR=***aaaa*

Explanation

A program issued an ERR$ (abort) request for the run-ID indicated. The E-Code *cccc* is program-supplied and is subject to the documentation for that program. The E-Type *tttt* can be decoded by @FAC.

The CPA procedure number which requested the ERR$ is *pppp*. The request occurred at virtual address *aaaa* in the procedure *pppp*

UP Key-in: *dddddddd* **Already UP, Input Ignored**

Explanation

You attempted an UP command on a device named *dddddddd* (PDT name) that is already up.

Volume Not Mounted

Explanation

The volume requested cannot be found.

Operator Action

Use the FS,MS command to determine the current online volume names (VOL: *vvvv*).

Wait - Last Input Ignored

Explanation

At this time, the program in control cannot accept your input.

Operator Action

Wait for an SOE to appear and reenter the data.

Wait - System Busy - Last Input Ignored

Explanation

At this time, the system is unable to accept your input.

Operator Action

Wait for the system to become less busy and reenter the data. If the situation continues, abort suspect or unneeded runs by using the E command or, if necessary, the TERM command. Check for suspected mass storage or other port/line module problems. If all else fails, dump the DCP and reboot the system.

Warning: *nn* **Duplicated File(s) Marked 'DOWN'**

Explanation

You brought a volume online and the DCP/OS detected that if certain file names were enabled, there would be duplicate file names in the system. To avoid such ambiguity, the DCP/OS marked the duplicate file names as down on the volume being enabled.

Operator Action

Check the volume (by using @PRT,d volname) and note the names of the files flagged as down (those with a preceding asterisk). If you want to enable any of these files, thus overriding files currently enabled, use the @CAT command with the S or U option.

*** Workstation Input Timeout ***

Explanation

The DCP/OS timed out the requested 'message wait' for this workstation. An urgent message may be queued so the system only waits for input for a limited or predetermined time.

Appendix E
Common Line Module Identifiers

Table E–1 lists DCP Series line module types, their corresponding hardware IDs, mnemonics, short text, description, and microcode IDs. Line modules marked with an asterisk (*) are discontinued, while those marked with an exclamation mark (!) are not loadable.

Note: *There are additional customer and product specific microcode IDs that do not appear in this list.*

Table E–1. Common Line Module Identifiers

HID (hex)	Mnemonic	Short Text	Line Module Description	MCID
01	LM$DIDSK	IFDC-0	*I INT FLX DSK CNTLR (256 Bytes)	03
02	LM$BYTE	BIOC	*I BYTE I/O CNTLR	02
04	LM$SU39	SU39-P	HOST BLOCK MUX (SU00039 DIAG SECONDARY PORT)	58
05		SU39-S	HOST BLOCK MUX (SU00039 DIAG SECONDARY PORT)	
07	LM$BDBI	BDBI	*I BYTE INTERFACE	07
08	LM$IFDSK	IFDC-1	*I INT FLX DSK CNTLR (1024K Bytes)	08
09	LM$MDLM	MDLM	I SCSI (MDLM)	09
10	LM$PAR	PERIPH	*I 16-BIT PARALLEL CHANL (PU100)	10
11	LM$SU57	SU57	I HOST WORD CHANL (SU00057 32-BIT)	11
12	LM$ICC	ICC	*I INTERCOMPUTER CHANL (ICU 16-bit)	12

continued

Table E-1. Common Line Module Identifiers (cont.)

HID hex)	Mnemonic	Short Text	Line Module Description	MCID
14	LM$SU208	SU208P	HOST BLOCK MUX (SU00208 PRIMARY PORT)	48
15		SU208S	HOST BLOCK MUX (SU00208 DIAG SECONDARY PORT)	
19	LM$FEPI	FEPI	I* FEPI	19
38	LM$DIAL	DIALER	I AUTO DIAL	38
40	LM$ASYNC	ASYNC 1	I LOW COST ASYNC - TTY	40 * (see note)
44	LM$MLASN	ASYNC 4	MULTILINE ASYNC (4X1)	59
50	LM$SYNC	SYNC1	I LOW COST SYNC - Uniscope	50 * (see note)
54	LM$S4X1	SY/AS4	* 4X1 SYNC/ASYNC RS232 (64K BYTES) - Asynchronous - Uniscope - UDLC	 4D 4E 4F
55	LM$MLSYN LM$MLASYN	SY/AS4	4X1 SYNC/ASYNC RS232 or DUAL BUS RS232 4X - Asynchronous - Uniscope - UDLC	 4D 4E 4F
60	LM$RS449	MS449	* MED SPD RS449 (MSLLM) - Basic Asynchronous - Basic Synchronous - UDLC - 1100 FDX - REM1 - EBCDIC BSC - ASCII BSC	 20 24 28 30 38 40 42

continued

Table E-1. Common Line Module Identifiers (cont.)

HID (hex)	Mnemonic	Short Text	Line Module Description	MCID
61	LM$X21	MSX21	MED SPD X.21 or DUAL BUS X.21 MS - Basic Synchronous - UDLC (switched) - UDLC (leased line) - 1100 FDX - EBCDIC BSC - ASCII BSC	 25 29 55 31 41 43
62	LM$X20	MSX20	MED SPD X.20 or DUAL BUS X.20 MS - Basic Asynchronous	21
63	LM$TRNAP	MS232A	MED SPD RS232 or DUAL BUS RS232 MS - Basic Asynchronous	20
64	LM$DCSS	DCSS	DCSS 250K BPS CX	70
65	LM$TP	TWPAIR	TWISTED PAIR	27
67	LM$RS232	MS232M	MED SPD RS232 or DUAL BUS RS232 MS - Basic Asynchronous - Basic Synchronous - UDLC - 1100 FDX - REM1 - EBCDIC BSC - ASCII BSC	 20 24 28 30 38 40 42
6B	LM$MSTWX	MSTWX	* MSLLM (TWX) - Basic Asynchronous	22
6F	LM$MSNC	MNOCBL	MED SPD RS232 or DUAL BUS RS232 MS (Cable not connected)	-

continued

HID (hex)	Mnemonic	Short Text	Line Module Description	MCID
70	LM$HS449	HS449	HIGH SPD LOADBL (HSLLM) - Basic Synchronous - UDLC - 1100 FDX - REM1 - EBCDIC BSC - ASCII BSC	 26 2A 32 39 44 45
71	LM$HSX21	HSX21	HIGH SPD LOADBL (X.21) - UDLC - 1100 FDX	 2B 33
7D	LM$HS303	HS303	HIGH SPD (BELL 303) - Basic Asynchronous - UDLC - 1100 FDX - REM1 - EBCDIC BSC - ASCII BSC	 26 2A 32 39 44 45
7E	LM$HSV35	HSV35	HIGH SPD V.35 - Basic Synchronous - UDLC - 1100 FDX - REM1 - EBCDIC BSC - ASCII BSC	 26 2A 32 39 44 45
7F		HNOCBL	* HIGH SPD (Cable not connected)	-
A0	LM$8023	802.3	802.3 ETHERNET LAN	80
A1	LM$IHS2M	I20-HS	4X1 HS T1/E1 (ILM20-HS)	A1
A2	LM$I40TR	I40-TR	DUAL BUS TOKEN RING (ILM40-TR)	A2
A3	LM$I40FD	I40-FD	DUAL BUS FDDI (ILM40-FD)	A3
A4	LM$HSBMX	SU212	HOST BLOCK MUX (SU00212)	84
A5	LM$I40EN	I40-EN	DUAL BUS ETHERNET (ILM40-EN)	A5
A8	LM$S8X1 LM$IRS232	I20-8B	8X1 ASYNC RS232 (ILM20-8B) - Asynchronous - SDLC - AUC	 5D A8 A8

Microcode ID 00 = Line module microcode not loaded

Microcode ID EF = MIcroprogram was loaded but requires an end-of-function (EOF - 4 bytes of 00 and 1 byte of FF) to operate. The ROM still has control.

Microcode IDs
F0 through FF = An error was detected during the microprogram load and the line module is not operational.

Both of these line modules return a hardware ID of 50 following either power-on or master-clear. After they are loaded with operating parameters, they return an ID of 50 if the parameters are sychronous and an ID of 40 if the parameters are asynchronous.

Glossary

B

batch mode
The mode in which programs are started in background. A system operating in batch mode executes programs serially.

boot
To load the microcode and operating system into the random access memory of a computer.

bootstrap
The software program that loads the operating system.

C

cache
The banks of memory used to temporarily store data read from or written to disk so that future READ requests may be done by copying from the cache space rather than reading it from the disk again. Memory access is much faster from cache space than from disk space.

CLC
See communications line controller.

CMS 1100
See Communications Management System.

communications line controller (CLC)
A processor in a DCP that manages the flow of data between the communications processor and a set of line modules. *See also* input/output processor.

Communications Management System (CMS 1100)
The software that manages all data communications into and out of OS 1100 host computers. CMS 1100 provides an interface between the OS 1100 and either the Telcon/DCP or the GCS/CTMC network.

communications processor architecture (CPA)
The model for Distributed Communications Processors (DCPs). CPA defines relationships among DCP hardware and software components.

console mode
An environment in which the experienced user can control the system.

CPA
See communications processor architecture.

D

DCA
See Distributed Communications Architecture.

DCP
See Distributed Communications Processor.

DCP/OS
See Distributed Communications Processor Operating System.

DCP/OS console
An operator window through which to converse with the DCP/OS.

DCP/OS utilities
A suite of utility programs for file manipulation, DCP/OS configuration, and program building. A disk utility program is included.

DCP/OS virtual workstation (VWS)
A Telcon-owned workstation that has a Telcon session to the target DCP/OS. It is a virtual workstation to the target DCP/OS.

DCP/OS workstation
A workstation that is owned and operated by the DCP/OS, which has the PP chain for the workstation.

debug
To locate and correct errors in computer programs.

debug mode
An operating condition that provides commands that enable the user to correct and restart programs under development.

demand mode
The mode of operation in which a terminal operator can enter a job stream statement-by-statement and have each statement transacted immediately (on demand).

dictionary
A list of CPA entities in a program. The dictionary is produced by the @DCPBUILD or @BUILD utility.

Distributed Communications Architecture (DCA)
A Unisys proprietary network architecture and set of communications protocols based on the seven-layer Reference Model for Open Systems Interconnection. DCA supports the protocols required for several different network environments to interoperate. The main differences between DCA and OSI architecture are in protocol implementation. DCA software implementations allow integration and concurrent operation of appropriate protocol modules or protocol conversion software that provide functions required in OSI, TCP/IP, and other network environments.

Distributed Communications Processor (DCP)
A special-purpose computer designed exclusively for communications applications. The DCP is used as a front-end processor for OS 1100 computers, or to interconnect networks of OS 1100 computers and other machines. Depending on how it is configured, a DCP can function as a remote concentrator, a message-switch (or nodal) processor, or a front-end processor. DCPs are available in several models.

Distributed Communications Processor Operating System (DCP/OS)
A multi-programming/multi-tasking/multi-processing operating system for DCPs. DCP/OS supports architectural entities defined by CPA, and controls all DCP hardware operations. The DCP/OS executes programs that are installed from diskettes or that are initially downloaded to the DCP from an OS 1100 system. DCP/OS is released in absolute format as part of the Telcon release tape.

download
To transfer data from a large system to a smaller system. In the OS 1100 system, a download operation usually refers to transferring DCP/OS files and TELCON configuration elements to a DCP.

dump
To transfer DCP memory contents to a local disk or to a host computer.

F

FEP
See front-end processor.

file structure
The arrangement and structure of data in a file, including the sequence and size of the file's components.

front-end processor (FEP)
A communications computer (usually a DCP) associated with a host computer. It may perform line control, message handling, code conversion, error control and application functions such as control and operation of terminals. The function of the FEP is to offload the communications-related functions from the host.

full-duplex
A function of hardware whereby both ends of a communications circuit can transmit and receive data simultaneously across a serial or parallel communications link. *Compare with* two-way simultaneous, a function of software.

H

half-duplex
A function of hardware, whereby both ends of a communications circuit can transmit and receive data, though not simultaneously.

host computer
A computing system attached to a data transmission facility that executes application or system programs.

I

interface
(1) Generically, the point or set of points at which two autonomous entities establish contact. (2) A connection point at which a set of rules governs the exchange of data and controls the information between two adjacent architectural levels. These levels may be software or hardware areas. For example, in Information Services (IS) program-to-program services, the programming interface is the dialog between a user-written program and IPC. In IS command interfaces, the interface is the dialog between a shell script, C program, or operator and IS. DSS products and IS products provide program-callable interfaces and user interfaces.

input/output processor (IOP)
An independent hardware processor (within a DCP) that implements virtual port processors (PPs), which manage line modules. IOPs can support up to 16 different interfaces, which provide a powerful programmable communications channel. The IOP provides the line handling protocol required by a particular type or class. Previously, IOPs were referred to as communications line controllers (CLCs).

I/O
Abbreviation for input/output.

IOM
I/O module in a DCP. The physical rack containing up to three IOPs (I/O processors).

IOP
See input/output processor.

J

JCS

See job control statement

job control statement (JCS)
A JCL instruction that identifies a specific job and describes its requirements to the operating system.

L

line module (LM)
The hardware in the DCP that terminates and controls data transmission of serial communications lines, host channel connections, and peripheral connections.

LM
See line module.

load path
The instructions for downloading and booting the DCP operating system. Usually set by an operator using DCP control panel switches.

load path table (LPT)
A DCP/OS table that contains the details of each configured load path, including the boot type, flags, source, and destination of a boot or dump.

load path table entry
An entry that defines the boot type, flags, source, or destination of a boot or dump.

load switches
The switches on a DCP used to initiate the type of boot action you want.

local load
To boot the microcode and the DCP/OS from a local disk or diskette. A local load is usually executed after an initial download installation when there is a DCP/OS problem or the DCP has been powered off.

LPT
See load path table.

M

maintenance panel
An equipment component, usually containing control switches and indicators, through which a maintenance engineer services the equipment.

mass storage
The ancillary memory storage media, usually disk subsystems.

modem
An acronym for MOdulator-DEModulator, a device that converts serial digital data from a transmitting terminal to an analog signal suitable for transmission over a telephone line. The modem then reconverts the signal to serial digital data for acceptance by a receiving terminal. *Same as* data set.

module library file (MLF)
An omnibus element created with the @DCPAPP utility.MLFs which are brought together with @BUILD or @DCPBUILD to create an absolute element which is executable in the DC/OS environment.

N

network
A group of hardware and software components that are physically and logically linked, and that interact according to established protocols. Network functions are determined by the types of cooperating application systems within the network.

NMS console
An operator window through which to converse with the NMS screen.

P

path number
An index into the load path table.

PDT
See physical device table.

physical device table (PDT)
A table that contains the name and device-dependent details of every device configured for DCP/OS use.

port
The unique physical address of each line module installed in a DCP. *See also* port number.

port number
The physical address of a DCP line module. For example, the port number may be specified in a configuration source file for a cross-channel load or a communications line download through a specific line module.

port path
A path that specifies the I/O port to be used when booting the DCP/OS.

program
A series of instructions in a form acceptable to a computer. In the context of run processing, a program is an absolute element to be executed. A program may be a system utility or a user program.

program load
A process that initiates a load and is faster than a system reset. When dumping files, use the program load button, since the system reset switch erases all memory.

project ID
A name or number that classifies a run for file access and accounting. The project ID is also used as the qualifier for file names when no qualifier is specified.

protocol
A general term for the predefined sequence of requests and responses by which units in a network coordinate control operations, data transfer operations, and other operations between the units.

Q

qualifier
On DCP systems, a one-to-six-character prefix used with a character file name (maximum 8 characters) to designate a file. The qualifier makes the file name unique. An asterisk separates the qualifier from the file name. If you do not provide a qualifier, the system uses your project identifier.

R

RAM disk
DCP memory set aside to simulate mass storage. A RAM disk contains files and may be brought up (UP), taken down (DN), or the memory may be freed (FR) using console mode commands. RAM disk differs from cache in that the data in RAM disk does not exist anyplace other than in DCP memory. In the case of a system crash, all files in RAM disk are lost.

remote concentrator (RC)
A communications computer (usually a DCP) with a direct connection to only one termination system (TS) of the two TSs that make up the end points of a session. This enables multiplexed communications between many low speed synchronous or asynchronous lines, and one or more high speed, usually synchronous, lines. The remote concentrator can be polled by a computing system and can in turn poll terminals. For example, if several terminals are directly connected to a node (the RC), messages from the RC can go to another node (an FEP), and then into a host.

resident
Code that resides completely in memory.

run
A sequence of statements that typically invokes programs.

run name
A character string that identifies a run to the DCP/OS.

runstream
A sequence of linked Exec control language (ECL) statements that form a self-contained unit of work.

S

spool file
A file that contains the output generated from a runstream.

status display
A display of information on your terminal.

SYSCFG
The system configuration file name that contains the physical device table (PDT) and the load path table (LPT).

system reset
A procedure that erases all memory in the DCP and initiates the loading of microcode and DCP/OS. Always use system reset if you are booting a new level of microcode or bootstrap, or if you want to load using a port path number.

system volume
The volume from which DCP/OS is loaded when the system is booted. There is no designated system volume if DCP/OS is memory resident.

T

Telcon virtual workstation
A workstation that is owned and operated by Telcon, which has the PP chain for the workstation.

transient
Code that is loaded into memory from the disk.

triple-E error
An internal error code in the range X'EEEx'. This implies a range of sixteen hexadecimal values: EEE0 to EEEF.

U

Universal Terminal System (UTS)
A family of Unisys interactive display terminals that uses the UNISCOPE line protocol.

UTS
See Universal Terminal System.

V

volume space table (VST)
A table on a mass storage device that contains a freespace list.

W

word
(1) The smallest addressable segment of main storage. (2) The smallest addressable unit of data on an OS 1100 system, consisting of 36 bits, and capable of being stored in a single main storage location. One word of data can represent 4 or 6 bytes, depending on whether ASCII or Fieldata format is used.

workstation (WS)
Either a Universal Terminal System (UTS) or a UTS-emulated terminal that is physically connected to a port (PP).

Bibliography

DCP Series Distributed Communications Processor Operating System (DCP/OS) Programming Reference Manual (7431 6894). Unisys Corporation. Previous document number: UP-11540.

DCP Series DCP/5 Installation and Operations Reference Manual (UP-14133). Unisys Corporation.

DCP Series Communications Processor Architecture Programming Reference Manual (7431 5805). Unisys Corporation.

DCP Series Telcon Operations Guide (7831 5785). Unisys Corporation.

DCP Series Telcon Operations Reference Manual (7831 5728). Unisys Corporation.

OS 1100 Communications Management System (CMS 1100) Operations Reference Manual (7831 5694). Unisys Corporation.

OS 1100 Distributed Communications Architecture (DCA) Technical Overview (UP-9676 Rev. 1). Unisys Corporation.

OS 1100 Exec System Software Executive Control Language (ECL) Operations and Programming Reference Manual (7830 7949). Unisys Corporation.

OS 1100 Exec System Software Installation and Configuration Guide (7830 7915). Unisys Corporation.

OS 1100 Text Editor Operations Reference Manual (UP-8723). Unisys Corporation.

OS 1100/DCP Series Communications Delivery Software Configuration Guide (7831 5678). Unisys Corporation.

OS 1100/DCP Series Communications Delivery Software Configuration Reference Manual (7831 5686). Unisys Corporation.

OS 1100/DCP Series Communications Delivery Software Installation Guide (7831 5645). Unisys Corporation.

Index

A

B

C

E

H

I

J

L

M

N

O

Q

R

S

U

V

W

X