Worldwide Response Center

HP 3000 APPLICATION NOTE # 91

Using The Port Structure
Under MPE/XL

HEWLETT PACKARD

October 01, 1991

Document P/N 5960-6659

RESPONSE CENTER APPLICATION NOTES

HP 3000 APPLICATION NOTES *are published by the Worldwide Response Center and are distributed with the Software Status Bulletin. These notes address topics where the volume of calls received at the Center indicates a need for addition to or consolidation of information available through HP support services.*

Following this publication you will find a list of previously published notes and a Reader Comment Sheet. You may use the Reader Comment Sheet to comment on the note, suggest improvements or future topics, or to order back issues. We encourage you to return this form; we'd like to hear from you.

HP 3000 Application Note # 91

USING THE PORT STRUCTURE UNDER MPE/XL

Introduction

Ever since it was created, the MPE/XL operating system has always used a specific concept to support the transmission of data between the processes and the process synchronization of the operating system itself.

This concept is known as the PORT FACILITY.

The purpose of this Application Note is to give a general outline of this new system functionality as the MPE/XL release 2.1, in conjunction with the Architected Interface, now allows the users to access these structures.

The following considerations shall be discussed hereunder:

What is a PORT? How are the PORTs used? What is a SUBQUEUE? What is a PORT SERVER? What are the advantages of using the Architected Interface in connection with the PORTs?

WHAT IS A PORT?

A port is a data structure dedicated to the reception of messages. Under MPE/XL, the only possibility for a system process A to communicate with a system process B consists in the former sending a message to the PORT and process B going to read it on the said PORT.

If needed, process B may come and wait on this PORT until a message comes in, in which case it remains in "standby" state. The incoming message from process A will "awake" B which immediately reads the message made available on the PORT.

For those users accustomed to MPE/V, this inter-process communication feature was formerly supported by the "eXtra-Data Segments", and intrinsics SENDMAIL, RECEIVEMAIL. Note that these concepts are maintained under MPE/XL, albeit only in compatible mode.

In this paper, the following structure will be used to represent a PORT:

```
 --------------------------------
|          |                     |
 -----------                     |
|          |      P O R T        |
 -----------                     |
|          |                     |
 --------------------------------
```

There are three different types of PORTs:

1. MESSAGE PORTs: the most widely used, for they allow "plain" inter-process message transmission. In other words, a message may consist of a set of characters in a format that can be understood both by the "writer" and the "reader" processes.
2. SIGNAL PORTs: these have the same characteristics as the above Message Ports except that the message size is cut to zero byte. Use of the SIGNAL PORT, therefore, will be confined to event triggering and inter-process synchronization.
3. SEMAPHORE PORTs: in many ways similar to the SIGNAL PORTs, they were used in the initial versions of MPE/XL. They have now been totally abandoned for the SIGNAL PORTs.

Before we go any further, let us study briefly a concrete example of how a PORT is used by MPE/XL.

A user process is running a FREAD on a file which internally results in this process writing a message onto a PORT of the IO system module. The message will contain information to answer the following questions: Where exactly on the disk should you read from? What length of disk should be read? In what area of the memory should the data read be stored? Once these questions have been sent to the IO system, the user process issues a message reception request on a PORT called IO_COMPLETION_PORT.

As can be assumed from its name, this is the port to which the IO module is going to write its message. This message will contain the IO report. The user process then reads the message, and if the report shows that the IO has been successful, the process may continue to run.

This very simple example summarizes the various functions of the PORTs: data transfers between the processes and process synchronization.

HOW ARE THE PORTs USED?

When a message is sent to a port, it is stored there in a chained FIFO (First-In First-Out)-type list. This means the messages will be read by the process waiting for reception on this PORT, in their order of arrival. Therefore, it is possible to have several messages waiting to be processed on a given PORT.

A process, however, can send only one message at a time to a PORT.

If several messages are to be sent, the process must invoke the sending procedure as many times as required.

Up to now, we have examined the PORT as a global concept. Let us now turn to the notion of SUBQUEUE.

WHAT IS A SUBQUEUE?

Each PORT is divided into a set of SUBQUEUES (up to 32 per PORT), which actually represent the various chained lists that are attached to a given PORT.

So, this SUBQUEUE concept will be used to assign priorities to the messages queued on this PORT, with SUBQUEUE 0 always having a higher priority than SUBQUEUE 31.

When a message is to be sent to a PORT, the process may do so without specifying the SUBQUEUE, in which case it defaults to SUBQUEUE 0. However, it may also choose to specify a specific SUBQUEUE.

On another hand, a process may wait globally on a PORT, that is it will receive the first message of the highest priority SUBQUEUE. A process may also wait on a given SUBQUEUE of a specific PORT.

Last of all, a SUBQUEUE may be either in ENABLE or DISABLE mode. If it is ENABLEd, it is possible to read the messages reaching this PORT, but if it is DISABLEd, the messages received on this SUBQUEUE cannot be read, unless the relevant SUBQUEUE number is specified.

Example:

This typical example should help you better understand the concepts of SUBQUEUE, ENABLE, and DISABLE.

Assume a PORT with three SUBQUEUES (0, 1, 2), where SUBQUEUES 0 and 2 are ENABLEd, and SUBQUEUE 1 is DISABLEd. A number of messages are sent to this port on different SUBQUEUES, as shown in the diagram below. Each message here contains the sending order number on this PORT. So we shall issue a number of reception requests to clearly understand in what order the messages are read.

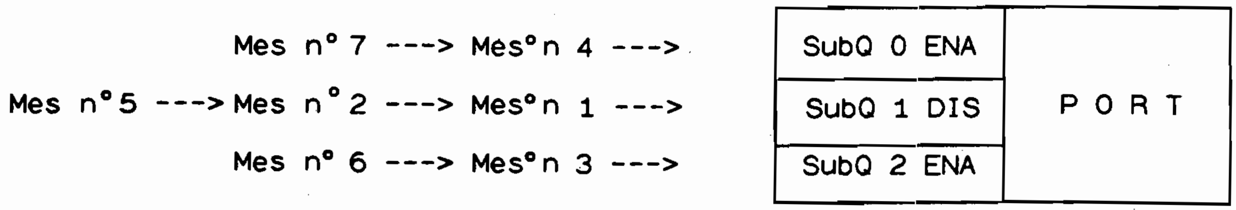

Figure 1-1.

Request No.1: read without specifying a SUBQUEUE ==> Mess 4. In this case, a message is received from the highest priority ENABLEd SUBQUEUE (0 here).

Request No.2: read SUBQUEUE No. 1 ==> Mess 1. Although it is DISABLEd (this may be assumed from the fact that the SUBQUEUE for reception has been specified), the first message present on SUBQUEUE 1 is received.

Request No.3: read from SUBQUEUE No.0 ==> Mess 7. No comments ... (Message 4 has already been read)

Request No.4: read without specifying any SUBQUEUE ==> Mess 3. There is no message left in SUBQUEUE 0, SUBQUEUE 1 is DISABLEd, the first message available on the next SUBQUEUE in terms of priority ranking is No.3.

After the concepts of PORT, SUBQUEUE, priority in the SUBQUEUES, we shall now turn to the concept of PORT SERVER.

WHAT IS A PORT SERVER?

A SERVER is a specific section of code which knows where to read and how to process a message on a given PORT. Thus, the function of the PORT is both to store the messages, and also to invoke the SERVER as soon as a message has arrived.

It is mandatory that a SERVER should be non re-entrant, in other words, if a message has just been received on a PORT, the SERVER will run. If, while it is executing, the SERVER is interrupted by the DISPATCHER (pre-emption or timesliced), and another message is received on the PORT, the SERVER is not invoked again. This provides for consistency with the above- described notion of message priority. On completing processing of a message, however, a SERVER is immediately re-invoked if a message has arrived while the former message was being processed, and so on until the PORT is empty.

The various types of SERVERs are described below:

PROCEDURE SERVER: in this case, the SERVER is a procedure to be run on the stack of the write process. The advantage of this procedure is that running it does not require loading any specific process. This is why this SERVER is particularly effective in terms of performance.

PROCESS SERVER: here the SERVER is a process with its whole operating environment, that is the stack, code and system structures required for its execution. The drawback of the SERVER procedures is that when you start them up, you have to switch contexts, which involves substantial overhead.

The PROCESS SERVER of a PORT is defined at PORT creation, that is as an intrinsic feature of this PORT.

ANY PROCESS SERVER: the characteristic of a PORT defined with ANY PROCESS SERVER is that it may be served by any process which has made a message reception request. In this case, a unique process may wait on a PORT. If a second process attempts to wait on the same PORT, an error is returned.

WHAT ARE THE ADVANTAGES OF USING THE ARCHITECTED INTERFACE IN CONNECTION WITH THE PORTS?

After reviewing the concept of PORT, let us now discuss the contribution of the ARCHITECTED INTERFACE (available for MPE/XL 2.1).

The Architected Interface allows you to integrate the PORT concept in your programs with all the related facilities offered in terms of process synchronization. This interface allows you to use all the above functions with the exception of the PROCEDURE SERVER.

Four procedures are available:

AIFPORTOPEN: creates a PORT, if it does not exist yet, or opens it if it already exists (like a file, a PORT must be open to be used). A PORT gets a name at creation time, this name should be used for all references to the PORT. The PORT access is protected by a password.

AIFPORTCLOSE: closes a PORT, or deletes it.

AIFPORTSEND: this procedure is used to send a message to a PORT. The maximum length of the message is 256 bytes.

When writing a message to a PORT, its priority must be defined; in fact, this priority matches the priority of the SUBQUEUE to which it belongs.

Moreover, when the user is sending a message, he is allowed to specify a TIMEOUT in number of seconds. If different from Zero (unlimited wait), this timeout is equal to the time a message remains on a PORT before it is deleted if no process has come to read it. If the TIMEOUT is greater than or equal to 0, the write process is locked in until this message is read or deleted. Otherwise, if the TIMEOUT is negative, the message is written in NONWAIT mode, that is the write process is not locked in until the message has been read or deleted.

AIFPORTRECEIVE: this procedure allows reading a message and setting a process to waiting for reception on a PORT. A TIMEOUT may be set for the waiting, a PRIORITY_MASK which actually defines the ENABLE/DISABLE SUBQUEUES.

CONCLUSION

Message transmission, process synchronization, task priority management ... all of these new functions can now be used by your programs in NATIVE mode from the moment you acquire the Architected Interface.

Published Application Notes

HP 3000

Following is a list of the Application Notes published to date. If you would like to order single copies of back issues please use the *Request Form* attached and indicate the number(s) of the note(s) you need, and the part number(s).

Note #	Part Number	Topic
1	5958-5824	Printer Configuration Guide - Version 1
2	5960-2841	Terminal types for HP 3000 HPIB Computers - Version 1
3	5960-2842	Plotter Configuration Guide
4	5960-2843	Printer Configuration Guide - Version 2
5	5960-2844	MPE System Logfile Record Formats
6	5960-2845	Stack Operation
7	5960-2846	COBOL II/3000 Programs: Tracing Illegal Data
8	5960-2847	KSAM Topics: COBOL's Index I/O: File Data Integrity
9	5960-2848	Port Failures, Terminal Hangs, TERMDSM
10	5960-2849	Serial Printers - Configuration, Cabling, Muxes
11	5960-2850	System Configuration or System Table Related Errors
12	5960-2851	Pascal 3000 - Using Dynamic Variables
13	5960-2852	Terminal Types for HP 3000 HPIB Computers - Version 2
14	5960-2853	Laser Printers - A Software and Hardware Overview
15	5960-2854	FORTRAN Language Considerations - A Guide to Common Problems
16	5960-2855	IMAGE: Updating to TurboIMAGE & Improving Database Loads
17	5960-2856	Optimizing VPLUS Utilization
18	5960-2857	The Case of the Suspect Track for 792X Disc Drives
19	5960-2858	Stack Overflows: Causes & Cures for COBOL II Programs
20	5960-2859	Output Spooling
21	5960-2860	COBOLII and MPE Intrinsics
22	5960-2861	Asynchronous Modems

HP 3000 (continued)

Note #	Part Number	Topic
23	5960-2862	VFC Files
24	5960-2863	Private Volumes
25	5960-2864	TurboIMAGE: Transaction Logging
26	5960-2865	HP 2680A, 2688A Error Trailers
27	5960-2866	HP Trend: An Installation and Problem Solving Guide
28	5960-2867	The Startup State Configurator
29	5960-2868	A Programmer's Guide to VPLUS 3000
30	5960-2869	Disc Cache
31	5960-2870	Calling the CREATEPROCESS Intrinsic
32	5960-2871	Configuring Terminal Buffers
33	5960-2872	Printer Configuration Guide - Version 3
34A	5960-2873	RIN Management (Using COBOLII Examples) (A)
34B	5960-2874	Process Handling (Using COBOLII Examples) (B)
35	5960-2875	HPDESK IV (Script files, FSC, and Installation Considerations)
34C	5960-2876	Extra Data Segments (Using COBOLII Examples) (C)
36	5960-2877	Tips for the DESK IV Administrators
37	5960-2878	AUTOINST: Trouble-free Updates
38	5960-2879	Store/Restore Errors
39	5960-2880	MRJE Emulates a HASP Workstation
40	5960-2881	HP 250 / 260 to HP 3000 Communications Guidelines
41	5960-2882	MPE File Label Revealed
42	5960-2883	System Interrupts
43	5960-2884	Run Time Aborts
44	5960-2885	HPPA Patching Conventions for HP3000 900 Series Processors - Version 1
45	5960-2886	Vplus & Multiplexers
46	5960-2887	Setting Up an HPDesk HPTelex for the First Time
47	5960-2900	Customizing Database Data Items & Changing Passwords in JCL Files
48	5959-9215	Printer Configuration - Version 4
49	5959-9227	Configuring DATACOMM Products Into MPE
50	5959-9228	VFC's for Serial Printers

HP 3000 (continued)

Note #	Part Number	Topic
51	5959-9237	Terminal Types for the HP 3000 HPIB Computers
52	5959-9242	Configuring MRJE
53	5959-9245	Using Special Characters on the 700/9x Series Terminals
54	5959-9251	Improving Database Performance
55	5959-9258	Customized Message Catalogs and Help Facilities
56	5959-9266	BRW Tips for Beginners
57	5959-9270	Configuring the HP 2334A Plus & HP 2335A As a Statistical Multiplexer
58	5959-9274	HPPA Pathing Conventions for HP3000 900 Series Processors - Version 2
59	5959-9289	HP 2334A and HP 2334A Configuration Recipes
60	5959-9301	TurboIMAGE's I-FILES and J-FILES
61	5959-7385	HPDeskManager - Looking Behind the Scenes
62	5959-7803	Setting Up a System Dictionary
63	5959-7834	Configuring Telesupport Modems for MPE V/E Systems
64	5960-1816	Finding Solutions in HP SupportLine
65	5960-1817	Using the Electronic Call Feature of HP SupportLine
66	5960-1818	Using the Feedback Feature of HP SupportLine
67	5960-1819	Printing Documents from HP SupportLine
68	5960-1820	HP SupportLine Commands
69	5960-2901	Nonsystem Volume Sets and the Migration of Private Volumes to an S9000 HP 3000
70	5960-2907	Modem Links for Remote Console and Standard DTC Connection s on Commercial XL HPPA Systems
71	5960-2918	Asynchronous Cabling
72	5960-2919	BRW Tips and Tricks
73	5960-2998	SNA NRJE Configuration
74	5960-2999	SNA IMF Configuration
75	5060-3000	XL NRJE Configuration

HP 3000 (continued) (continued)

Note #	Part Number	Topic
76	5960-4301	XL IMF Configuration
77	5960-4302	Calling the BRW Intrinsics
78	5960-4303	PUB.SYS What Is Behind It?
79	5960-4625	Conquest of Disc Space
80	5960-4633	Looking Behind the Scenes of Resource Sharing
81	5960-4637	MPE/XL System Interrupt Recovery Procedures
82	5960-4347	Private Volumes
83	5960-4396	Serial Printer Configuration
84	5960-4334	How to Migrate FORTRAN Programs to Newer Compilers and XL Hardware
85	5960-4335	The Optimization of Programs in MPE/XL
86	5960-4643	IBM Labeled Tapes Questions and Answers
87	5960-4666	Image Logging for HP Financial Accounting Databases
88	5960-4672	Native Mode Spooler Questions and Answers
89	5960-4673	AUTOINST/XL Questions and Answers
90	5960-4701	The New Spooler
91	5960-6659	Using the Port Structure Under MPE/XL

HP 3000 Application Note Request

☐ Please send me the HP 3000 Application Notes listed below. To order an Application Note just fill in the Application Note number and the part no. in the space provided below. (Use the Published Application Notes form)

AN __/__ 59 __/__-__/__/__/__ AN __/__ 59 __/__-__/__/__/__

AN __/__ 59 __/__-__/__/__/__ AN __/__ 59 __/__-__/__/__/__

AN __/__ 59 __/__-__/__/__/__ AN __/__ 59 __/__-__/__/__/__

AN __/__ 59 __/__-__/__/__/__ AN __/__ 59 __/__-__/__/__/__

AN __/__ 59 __/__-__/__/__/__ AN __/__ 59 __/__-__/__/__/__

Business phone	Extension	Best time to call

Name ______________________ Title/Dept.

Company ______________________ Division

Mailing Address ______________________ Mail Stop/Bldg./Rm.

City ______________ State ______________ Zip Code

1515

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO. 809 CUPERTINO, CA

POSTAGE WILL BE PAID BY ADDRESSEE

HEWLETT-PACKARD
19310 PRUNERIDGE AVE
BLDG 49 AM
CUPERTINO CA 95014-9826

Customer Comment Card

Application Note No.: ________

Part No.: ________________

We welcome your evaluation of this Application Note. Your comments and suggestions will help us improve our publications. Attach additional pages if necessary.

Please circle the following Yes or No:

■ Is the information technically accurate?	Yes	No
■ Are instructions complete?	Yes	No
■ Are concepts and wording easy to understand?	Yes	No
■ Are the examples and pictures helpful?	Yes	No
■ Is the format of this note convenient in size, arrangement and readability?	Yes	No
■ Did you receive the published application notes requested in a timely manner?	Yes	No

Additional Comments and/or suggestions for future application notes: ________________

____

____

____

____

Please provide:

Name: ________________ Title: ________________

Company: ________________ Address: ________________

City: ____________ State: _____ Zip Code/Country ________________

Please send to :

HEWLETT PACKARD

No postage is required. Just remove this card, fold so that the pre-addressed label is on the outside, secure and mail.

Thank you for your assistance.

Printed in U.S.A. 1990

FOLD FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 95, MT. VIEW, CA

POSTAGE WILL BE PAID BY ADDRESSEE

Learning Products Manager
Hewlett-Packard Company
100 Mayfield Avenue
Mail Stop 37MA
Attention: George Enos
Mt. View, CA 94043

FOLD FOLD

** For HP Internal Reference Only **

Manufacturing Part Number

5960-6659

Customer Order Number

NONE

Printed in USA