

TO: Distribution
FROM: Robert S. Coren
DATE: April 23, 1973
SUBJECT: Special I/O Daemon Operations

This memo describes two special features available on the I/O Daemon which may be used to:

1. Create the I/O Daemon queues from scratch
2. Process special queues instead of the default ones

I. Creating I/O Daemon queues

Before an I/O Daemon can be run on a system, a variety of data bases must be created and initialized: in particular, 3 message segments to be used as request queues (named "io_daemon_n.ms", where n = 1,2,3), an "info" data segment (named "daemon_info") and a segment for saving current requests (named "daemon_save_seg"). These segments must all reside in a single directory (normally >daemon_dir_dir>io_daemon_dir). To establish these segments in a given directory the following command may be given (before the daemon is initialized):

```
daemon_init$init_info dir_name
```

where "dir_name" is the directory in which the segments are to be created (usually >daemon_dir_dir>io_daemon_dir).

In the case of a cold boot or if >daemon_dir_dir is destroyed and it is unable to be retrieved, the initializer's "exec create_ccd" may be used to create the entire >daemon_dir_dir hierarchy including the io_daemon_dir and associated segments. It should not be necessary to execute the above command except to create special I/O Daemon queues (see part II below). For example, to create a set of I/O Daemon queues in >ldd>listings to be used for weekend processing, the following command could be issued:

```
daemon_init$init_info >ldd>listings
```

II Using Non-standard I/O Daemon Queues

In order to run an I/O Daemon using data bases previously created and initialized in a directory other than >daemon_dir_dir> io_daemon_dir, the command line:

```
daemon_init$test_lo dir_name
```

should be used, where "dir_name" is the full pathname of the desired directory. The I/O daemon and the dprint command will use this directory for the remainder of the process, or until the directory is reset by another call to daemon_init\$test_lo.

For users to place requests in a special set of I/O Daemon queues, the following command should be used:

```
dprint_$test dir_name
```

where "dir_name" is the full path name of the directory in which a set of I/O Daemon segments have been created as above. After this command has been issued, all further dprint/dpunch requests will be put in the special queues specified.

III Access Control on the I/O Daemon Queues

Unless altered, the "extended access" on the message segments will be set to "adros" for IO.SysDaemon.* and "aos" for " *.*.*": thus the I/O Daemon will have the ability to add, read, or delete any message in the queue, while any other user will be able to add messages and find out how many there are, but will only be permitted to read or delete messages which he himself added.

At MIT, the access to the standard queues has been modified to allow more of the "system" projects to have complete access to the queues and to allow everyone else to read all requests. The following is the "extended" access control lists for each of the standard queues:

```
adros *.SysDaemon.*
adros *.SysAdmin.*
adros *.SysMaint.*
adros *.Operator.*
aros  *.*.*
```

To modify the "extended" access control lists, the command "message_segment_setacl" ("mssa") can be used. For example, to allow members of the SysLib project, to have the ability to add, list and delete any request in the special queues created above in >ld>listings, the following command could be used:

```
mssa >ld>listings adros *.SysLib.* null *.*.*
```

Descriptions of the message segment commands may be found in the Systems Programmer's Supplement of the Multics Programmer's Manual. Descriptions of the list_daemon_requests (ldr) and cancel_daemon_requests (cdr) may be found in the Multics Programmer's Manual (draft copies attached).

Command
Development System
4/18/73

Name: list_daemon_requests, ldr

The list_daemon_requests command allows the user to obtain information about oprint and dpunch requests. Normally the user will be allowed information only concerning requests which he has made.

Usage

list_daemon_requests control_arg1 ... control_argn

- 1) control_arg1 is selected from the following list of control arguments and may appear anywhere on the command line:
- total, -tt indicates that the user wants only the total of the requests in the queue.
 - long, -lg indicates that all of the information pertaining to a request will be printed. If this option is omitted, only the full path name of the segment to be printed or punched will be printed.
 - queue n, -q n indicates which queue is to be searched. It must be followed by an integer specifying the number of the queue. If this option is omitted, the third priority queue is searched unless the -all option is provided. (See below.)
 - all, -a indicates that all priority queues are to be searched starting with the highest priority queue and ending with the lowest priority queue.

Note

The -total and -long control arguments are incompatible.

Examples

1) list_daemon_requests

Queue 3: 3 requests. 6 total requests.

```
>udd>Multics>Jones>dump>translate.list  
>udd>Multics>Jones>doc>ldr.runout  
>udd>Multics>Jones>Jones.profile
```

Page 2

2) list_daemon_requests -long -queue 1

Queue 1: 2 requests. 27 total requests.

Pathname: >uod>m>Day>foo.list
Type: print
Copies: 1
Delete: yes
For: Jones

Pathname: >ooc>Info>motd.info
Type: print
Copies: 3
Delete: no
To: 575 Tech Sq.

3) list_daemon_requests -total -all

Queue 1: 2 requests. 15 total requests.

Queue 2: 0 requests. 0 total requests.

Queue 3: 0 requests. 39 total requests.

Command
Development System
4/22/73

Name: cancel_daemon_request, cdr

The cancel_daemon_request command allows a user to delete a dprint or dpunch request which is no longer required. Normally the deletion can be made only by the user who originated the request.

Usage

cancel_daemon_request pname -control_args-

- 1) pname is the pathname of the segment to be printed or punched by this request. The pathname must be typed as it was given in the original request.
- 2) control_args are selected from the following list of control arguments and may appear anywhere on the command line:
 - queue *n*, -q *n* indicates which priority queue is to be searched. It must be followed by an integer specifying the number of the queue. If this option is omitted, the third priority queue is searched unless the -all option is provided. (See below.)
 - all, -a indicates that all priority queues are to be searched starting with the highest priority queue and ending with the lowest priority queue.
 - brief, -bf indicates that the message "Dprint (dpunch) of pname cancelled" is omitted.

Notes

The last request to print or punch a segment is deleted if there is more than one request for that segment in the same queue.

If the request refers to a segment which the I/O daemon is already processing, this command will not be effective in stopping the print or punch operation.

Example

```
cdr >udd>Multics>Jones>dump>translate.list
```

would delete the last request which the user had made in queue 3
to print or punch the segment >udd>Multics>Jones
>dump>translate.list.