

TO: Distribution
FROM: Gary C. Dixon
DATE: March 26, 1975
SUBJECT: enter_output_request, a replacement for dprint/dpunch

Attached is the description of a new command which is being proposed as a better interface to dprint than the dprint than dpunch commands. The command is enter_output_request (eor) and its associated entry points, print_output_defaults (pod), and change_output_defaults (cod). These three commands are renamed extensions of the commands proposed in MTB-002. The extensions are based upon the comments received in response to MTB-002 and based upon recent IO Daemon extensions. The command names were changed from those in MTB-002 (enter_daemon_request, print_daemon_defaults, and change_daemon_defaults) to be more descriptive of the function being performed.

Recalling from MTB-002, the major extension embodied in the new eor command allows each user to change the values of the default heading line and destination used for each output request. Further extensions proposed here separate the user-settable defaults for printing and punching, and allow default values for the request type (formerly device class), queue number, number of copies, output mode (-bf/-lg), deletion mode (-dl), and all of the other control options, as well as for the heading and destination.

In addition, the new command provides a -force (-fc) control argument which gives the IO Driver process read access to the file being output, in the event that the IO Driver which corresponds to the (user-supplied or default) request type does not have read access.

Finally, the new command accepts the star convention. As a safety check, it provides a user-settable limit which controls the maximum number of files matching a given star name which can be submitted without questioning the user. A new -list (-ls) control argument supports the use of the star convention by listing the entry names of files being submitted.

The main reason for creating a new command, instead of merely extending the dprint and dpunch commands, is to change the way in which control arguments are handled by the extended command. dprint and dpunch are two of the few MPM-documented

Multics Project Internal working documentation. Not to be reproduced or distributed outside the Multics Project.

commands which attach some meaning to the position of their keyword control arguments (i.e., the control arguments apply only to those segments whose names appear after the control arguments in the command). Attaching meaning to the position of control arguments violates Multics "System Programming Standards" for Argument Handling, Rule 6 (MPM Reference Guide, Section 2.5, page 11) which states:

- 6) In general, for the convenience of the user, command arguments should be order independent unless the order dependency serves a useful purpose (as in the `-ag` control argument of the `enter_abs_request` command).

The positional control arguments of `dprint/dpunch` do not serve such a useful purpose. Instead, because of their variance from standard, they often lead to user confusion. Indeed, I was confused the first time I used `dprint`, after just having read its MPM description, and many new MIT users have mentioned a similar confusion on first using the command. In addition, even now `dprint` does not handle the use of multiple queue numbers or multiple request types in the same command correctly. There is no need to handle such complexities in `eor`.

The `dprint` and `dpunch` commands should be retained (with all of their entry points) for compatibility purposes while users are changing over to the new commands. However, documentation of `dprint` and `dpunch` should be removed from the MPM, and the commands should become obsolete interfaces, eventually to be removed from the distributed system. Also, no support of any new features should be added to the commands.

The `list_daemon_requests` command should be renamed to `list_output_requests` (`lor`), and its calling sequence should be extended to include optional starnames identifying the requests to be listed. `cancel_daemon_request` should be renamed to `cancel_output_request` (`cor`), and its calling sequence should be extended to include one or more star names in any argument position which identify the request(s) to be cancelled. Because the extensions to the calling sequence remain compatible with existing calling sequences, the old command names could be retained as alternate entry points for compatibility purposes during a changeover period.

The rather arbitrary maximum of 4 copies per output request should be increased to a user-settable limit which defaults to a 4 copy maximum. There is no reason for building a 4 copy maximum into the heart of the IO Daemon when many installations may find such a low limit overly-protective and bothersome. I propose a new default control argument, `-max_copies` (`-mcp`), which a user can set with the `change_output_defaults` command. If this maximum is surpassed in any given `enter_output_request` command, the user

would be asked if he really wants that many copies. If he replies yes, then eor would submit a request with the specified number of copies. This is reasonable, since the user is, in fact, paying for the copies he has requested, and we have warned him of the potentially mistaken number of copies. -max_copies would be set at 4 initially to conform with the current limit, but each user could adjust this value to his own needs. The 4 copy maximum should be removed from dprint_.

The default values for eor should be stored in the user's profile so that they may be retained across process boundaries and act as permanent default values. In order to use the profile in a flexible way, a generalized profile management routine should be created. Until such an interface is available, the default values can be stored on a per process basis as internal static values.

If you have comments on this MTB, please send mail to me on the MIT Multics or eor your comments to my output bin, as follows:

```
mail comments GDixon PDO
```

or

```
dprint -ds GDixon comments
```

Suggestions as to how to further grade error messages into severity groups would be especially welcome.

(This page intentionally left blank)

```
-----  
| enter_output_request |  
|-----|
```

Command
03/28/75

Names: enter_output_request, eor

The enter_output_request command submits requests to print or punch files (segments or multisegment files) on the Multics line printer or card punch. The printing or punching of each file is controlled by the system through an IO Driver process which selects files from one of three queues in first-in/first-out order. Files are selected first from the highest-priority queue (queue 1) until this queue is empty, then from the next-priority queue (queue 2), and finally from the lowest-priority queue (queue 3).

Each printed listing of a file is preceded by a header page which includes: the path name of the file; the process group id of the submitter; a user-defined heading line and destination; and the date and time of printing. Each punched card deck is preceded by flip cards which include: the process group id of the submitter; the heading line; the date and time of punching; and the path name of the file.

Usage

enter_output_request paths -control_args-

1. paths are one or more path names which identify to be printed or punched. Relative path names may be used, and the final entry name of the path may be a star name. If the path name identifies a link, then the file linked to will be printed or punched.

2. control_args may be any of the following optional control arguments. Path names and control arguments may appear in the command in any order. Each control argument applies to all path names in the command. If one or more of these control arguments are not specified, then default values are used, as explained in the description of the change_output_defaults command below.

-print, -pr specifies that the files are to be printed. This is the default operation.

```
-----  
| enter_output_request |  
|-----|
```

Page 2

- punch, -pch specifies that the files are to be punched.
- queue n
-q n n specifies the number of the queue to which the request is submitted. It must be a number between 1 and 3.
- header heading
-he heading heading is the character string which is used as the heading line in the header which precedes the printed listing or punched card deck. Only the first 64 characters of the heading will be used.
- destination dest
-ds dest dest is the character string which is used as the destination in the header which precedes the printed listing. Only the first 12 characters of dest will be used.
- copy n
-cp n specifies that n copies of the file are to be printed or punched.
- request_type type
-rat type specifies the type of IO Driver which is to process the request. (See Note below.)
- form form form specifies the name of the print form or punched card to be used for the output. Only the first 8 characters of form will be used.
- name name
-nm name specifies that name is a path name which identifies a single file to be printed or punched, even though name may appear to be a star name, or a control argument.

The following control arguments are paired together. One control argument of a pair causes an action to occur, the other suppresses that action. Only one control argument from each pair may appear in any given command.

- `-delete, -dl` specifies that each file is to be deleted after being printed or punched.
- `-no_delete`
`-ndl` specifies that each file is not to be deleted after being printed or punched.
- `-long, -lg` prints the status of the queue to which the requests were submitted.
- `-brief, -bf` suppresses printing of the queue status.
- `-list, -ls` specifies that file names are to be printed on the user's terminal as files are submitted for printing or punching.
- `-no_list, -nls` suppresses printing of file names.
- `-notify, -nt` specifies that the user is to be notified by message when each file is printed or punched.
- `-no_notify`
`-nnt` suppresses user notification.
- `-force, -fc` specifies that, if the IO Driver which is to process the request does not have read access to the file, `enter_output_request` should try to give it read access without informing the user.
- `-no_force, -nfc` specifies that, if the IO Driver does not have read access to the file, `enter_output_request` should ask the user if the request should still be submitted. If the user answers yes, then the request is submitted to the queue. The user should then give the IO Driver read access to the file.

The following control arguments may only be used when printing a file. Those arguments below which are paired (e.g., `-end_page` and `-no_end_page`) may not both appear in the same command.

```
-----  
| enter_output_request |  
|-----|
```

Page 4

- line_length *n***
-ll *n* *n* specifies the maximum number of characters which may be printed on each output line. Lines of the file which are longer than *n* characters will be printed as several *n*-character lines.
- page_length *n***
-pl *n* *n* specifies the maximum number of lines which may be printed on each page. After *n* lines have been printed on a page, a newpage character will be output to start a newpage, unless the **-no_endpage** control argument has also been specified (see below).
- no_endpage, -nep** suppresses the end-of-page checking which normally prevents more than one page length of output lines from being printed on each page. This allows the output to be printed over page perforations. Pages skips are completely controlled by the newpage (NP) characters which appear in the file being printed.
- end_page, -ep** specifies that end-of-page checking is to be performed to prevent more than one page length of output lines from being printed on a page.
- single, -sg** specifies that newpage (NP) and vertical tab (VT) characters appearing in the file are to be treated as newline (NL) characters. This control argument does not affect the end-of-page checking.
- control, -ctl** specifies that newpage (NP) and vertical tab (VT) characters appearing in the segment of multisegment file are to be treated normally.
- edited, -ed** specifies that non-printable characters are to be omitted from the output.
- non_edited, -ned** specifies that non-printable characters are to be printed as octal escape sequences

(e.g., \006).

-indent n
-in n

n specifies the number of characters by which each line is to be indented from the left margin of the print form. The n indentation characters are subtracted from the total line length when determining how many characters may be printed on each line.

-top_label head
-tbl head

head is a character string which is placed at the top of each page of printed output. Only the first 120 characters of head are used. The heading line is subtracted from the total page length when determining how many lines may appear on a page.

-bottom_label foot
-tbl foot

foot is a character string which is placed at the bottom of each page of printed output. Only the first 120 characters of foot are used. The footing line is subtracted from the total page length when determining how many lines may appear on a page.

-label label
-tbl label

label is a character string which is placed at the top and bottom of each page of printed output.

-label "string"
is equivalent to:
-top_label "label" -bottom_label "label"

-access_label
-tbl

specifies that the access class of each file is to be used as the top and bottom labels on each page of printed output.

-no_access_label
-tbl

suppresses any access label specification.

```
-----  
|  
| enter_output_request |  
|  
-----
```

Page 6

The following control arguments may only be used when punching a file. If they are used, then the `-punch` control argument is assumed. Only one of the following control arguments may be used in any given command.

<code>-mcc_punch</code>	
<code>-mcc</code>	specifies that the file is to be punched, using the Multics character code.
<code>-7punch, -7pch</code>	specifies that the file is to be punched, using the 7-punch character code.
<code>-raw_punch</code>	
<code>-raw</code>	specifies that the file is to be punched as a stream of bits, without conversion.

Note

Each installation may support several different types of printed or punched output. For example, there may be print types for regular printed output, for output on unlined paper, for output on a remote printer, and for output with a special print chain. Requests to print or punch files are stored in queues associated with the print or punch type specified by the `-request_type` control argument. If no `-request_type` control argument is given, then the requests are stored in the queues associated with the default print or punch type.

Entry: `change_output_defaults, cod`

When one or more control arguments are omitted from an `enter_output_request` command, default values are used for the omitted control arguments. Different default values are applied to print and punch requests. The default control argument values are initialized in each process as shown in the table below.

The `change_output_defaults` command can be used to change the default values which `enter_output_request` uses. `change_output_defaults` is called to change one or more of the values. Thereafter, these new default values are used by `enter_output_request` in that process.

```
-----  
| enter_output_request |  
|-----|
```

03/28/75
Page 7

Usage

change_output_defaults control_args

1. control_args may be any of the control arguments described above for the enter_output_request command (excluding the -name control argument), or may be any of the following special control arguments. By default, the control arguments specify the default values used for print requests, unless one of the -punch, -mcc_punch, -7punch, or -raw_punch control arguments is used.

-max_copy *n*
-mcp *n*

n places an upper bound on number of copies which may be specified in a -copy control argument. If a -copy control argument specifies more than *n* copies, the user is warned that the number of copies requested exceeds the max_copy bound and is asked if the request should still be submitted.

-limit *n*
-li *n*

n specifies the maximum number of entries matching a star name which may be submitted for printing or punching without questioning the user. This limit acts as a safety check which guards against the accidental use of a star name matching a large number of entries not intended for output.

-severity *n*
-sv *n*

n specifies an error severity number which may be 1, 2 or 3. This causes enter_output_request to suppress error messages whose severity is less than *n*. Error message severities are described below.

```
-----  
| enter_output_request |  
|-----|
```

Output Defaults

The following table shows the initial default values for the control arguments described above. These values may be changed by using the `change_output_defaults` command.

Initial Default Control Argument Values for

<u>Printing</u>	<u>Punching</u>
-print	-punch
-queue 3	-queue 3
-header [user name]	-header [user name]
-destination [user project]	-destination [user project]
-copy 1	-copy 1
-request_type default	-request_type pun_dflt
-form lined	-form standard
-no_delete	-no_delete
-long	-long
-no_list	-no_list
-no_notify	-no_notify
-no_force	-no_force
-line_length 136	-mcc_punch
-page_length 60	
-end_page	
-control	
-edited	
-indent 0	
-top_label ""	
-bottom_label ""	
-label ""	
-no_access_label	
-max_copies 4	-max_copies 4
-limit 20	-limit 5
-severity 1	-severity 1

[user name] and [user project] refer to the results of the user active function, and represent the user's name and project respectively.

Error Message Severities

`enter_output_request` can diagnose many different kinds of errors. These errors are graded in severity as follows:

<u>Severity</u>	<u>Meaning</u>
1	Warning that an entry matched by a star name is a directory, a link to a directory, or a null link. The entry is ignored, and other entries matching the star name are submitted.
2	Errors that occur during the processing, with the exception of those listed as warnings above.

Entry: print_output_defaults, pod

This entry point prints the current default control argument values for printing or for punching.

Usage

print_output_defaults -control_args-

1. control_args may be any of the control arguments described above for change_output_defaults. None of these control arguments accept argument values when used with print_output_defaults. Instead, the control arguments specify which current default argument values should be printed. The default values which are printed apply to print requests unless one of the -punch, -mcc_punch, -7punch, or -raw_punch control arguments is used.

Note

If no control arguments are given for print_output_defaults, or if only -print or -punch is given, then all of the current default control argument values are printed. Otherwise, only those default values identified by the control arguments are printed.