

Published: 05/06/68

### Identification

Media Operator Command  
media  
W. R. Strickler

### Purpose

The media command is the service procedure associated with the media unsolicited-request function. It interrogates the Media Request Manager (BT.2.02) for definitions of requests for media handling, informs the media operator of any such requests, and reports back to the Media Request Manager after each request has been serviced.

### Discussion

The current implementation of the media command is intended only for Initial Multics. It provides for servicing of only one media request at a time; a request is relayed by the Media Request Manager to the media procedure, which prints a console message informing the media operator of the request; he performs the requested service and reports success or failure of service; then the media procedure informs the Media Request Manager that service of the request is terminated, either successfully or unsuccessfully.

In later versions of Multics the media operator should be allowed to service several requests simultaneously, and to indicate gradual progress in the handling of each request, identifying it by an index.

### Usage

In Initial Multics the media command is invoked by the operator at his console when he is ready to service media requests. After being informed of a request, he attempts to perform the service and then types "yes" if he was successful or "no" otherwise.

Ultimately op\_checker (BX.15.03) will call media automatically to inform the media operator of requests for media handling as soon as they are signaled by other processes.



The media procedure takes the following steps:

1. Call `hcs_$mrm_get_request`.
2. Check `cstat`. If `cstat=1`, call `write_out` (BY.4.02) to print a console message that `mrm` has been called erroneously, and return. If `cstat=2`, return. Otherwise:
3. Inform the operator of the request. The character strings `op`, `type`, `medium`, `device_type`, and `device`, returned by `mrm_get_request` are combined with a preliminary message and printed, by calls to `write_out`, on three lines:

```
"Media service requested is "||op
"   "||type||" "||medium
"   "||device_type||" "||device
```

The strings, `type` and `device_type`, are stripped of trailing blanks.

4. Call `write_out` to tell the operator to type "yes" or "no" when service is complete, and go into IO-wait by calling `read_in` (BY.4.02) for the operator's reply.
5. If the character string typed by the operator is "yes", set the last bit of the status argument of `hcs_$mrm_put_status` to "1"b and go to step 6. If the string is "no", set status to "0"b and go to step 6. Otherwise, go to step 4.
6. Set the `eos` argument of `mrm_put_status` to "1"b.
7. Call `hcs$mrm_put_status`.
8. Go back to step 1 for more media requests.