MCR10205
IMFT: configurable wait for the EOF reply
Revision 1.0

Eric Swenson

November 10, 2026

1 Problem Description

An IMFT output driver sends an object by running `backup_dump_` into the transport, then sends an EOF command and waits for the remote input driver's EOF reply, which says whether the object was loaded without errors (`imft_transmit_object_`). The remote driver sends that reply only after `backup_load_` has loaded everything. The wait is fixed at 5 minutes: the EOF command is written with `local_icri.timeout = 5 * ONE_MINUTE` and the same value is used for the reply read.

So the sender's 5 minutes must cover the time the receiver needs to take in and load whatever is still in transit when the sender finishes, plus `backup_load_`'s own time for the last entries. When that is longer, the sender reports

```
imft_transmit_object_: Attempt to read/write record timed out. Attempting to read from
    SYSTEM.
```

drops the connection and defers the request, and the request fails the same way on every retry. This was seen transferring a large subtree over dial-up (FNP) lines between two emulated systems, where the emulator let the sender run several minutes of data ahead of the receiver. (That emulator problem has been fixed separately.) A slow line, a heavily loaded receiver or a large final segment can still need more than 5 minutes, and a site has no way to allow for it.

2 Proposed Fix

A new keyword for the IMFT output driver's `args:` statement in the I/O daemon tables, alongside `sleep_time=` and `min_time_est_msg=`:

```
eof_reply_timeout=N
```

`N` is the number of minutes to wait for the reply to an EOF command, from 0 to 1440; 0 means wait as long as it takes. The default is 5, so a driver whose `args:` statement doesn't use the keyword behaves as before. The keyword is rejected for an input driver, as `auto_go=` is, since only output drivers send EOF commands. An out-of-range or non-numeric value stops driver initialization with

```
Value of eof_reply_timeout must be a number between 0 and 1440 minutes, not "VALUE"
```

Only the read of the EOF reply uses the new value. Writing the EOF command and every other control-record exchange keep their fixed timeouts. A timeout of 0 is already supported by `imft_io_` (no alarm is set), and the driver itself uses it when waiting for a remote system to start synchronizing. With 0, a remote system that stops responding without closing the connection leaves the driver waiting until the operator intervenes; a hangup or a closed network connection still ends the wait with an error.

Changes:

- `_imft_driver_info.incl.pl1`: new field `eof_reply_timeout fixed bin (35)` (seconds) after `min_time_to_log`. The structure is included only by `imft_driver_`, `imft_transmit_object_` and `imft_receive_object_`, all in `bound_imft_`, and lives in `imft_driver_`'s internal static, so its layout isn't shared with anything else.
- `imft_driver_`: parse `eof_reply_timeout=` after `min_time_est_msg=` and store it in seconds. The change is made to the installed version, which includes MCR10197 (MR12.9-1088).
- `imft_transmit_object_`: set `local_icri.timeout = imft_driver_info.eof_reply_timeout` before reading the EOF reply.
- `cy73.errata.info`: new "Information for MR12.9" section correcting `version=` and documenting the IMFT `args:` keywords missing from CY73-01, including `eof_reply_timeout=` (see "Documentation").

`imft_receive_object_` is recompiled only because of the include file; its code is unchanged.

The following `compare_ascii` output of the original and new `imft_driver_.pl1` shows the proposed changes:

```
Inserted in B:
B587          keyword_value = iodd_parse_$args ("eof_reply_timeout=", major_args);
B588          if keyword_value = "" then
B589               static.eof_reply_timeout = FIVE_MINUTES;
B590          else if static.input_driver then do;       /* only output drivers send EOF
    commands */
B591               call iodd_msg_ (ERROR, MASTER, 0, IMFT_DRIVER_,
B592                  "The ""eof_reply_timeout"" keyword may not be specified for an input
    driver.");
B593               go to RETURN_FROM_INIT;
B594          end;
B595          else do;                                   /* minutes; 0 => wait as long as
    it takes */
B596               static.eof_reply_timeout = cv_dec_check_ (keyword_value, code);
B597
B598               if code ^= 0 | static.eof_reply_timeout < 0 | static.eof_reply_timeout
    > 1440 then do;
B599                  call iodd_msg_ (ERROR, MASTER, error_table_$bad_conversion,
    IMFT_DRIVER_,
```

```
B600                      "Value of eof_reply_timeout must be a number between 0 and 1440
    minutes, not ""^a""",
B601                      keyword_value);
B602                   go to RETURN_FROM_INIT;
B603                end;
B604                static.eof_reply_timeout = static.eof_reply_timeout * ONE_MINUTE;
B605           end;
B606
Preceding:
A587           static.debug_mode, static.copy_data = "0"b; /* defaults to not debugging, of
    course */

Comparison finished: 1 difference, 20 lines.
```

The following `compare_ascii` output of the original and new `imft_transmit_object_.pl1` shows the proposed changes:

```
Inserted in B:
B482                 local_icri.timeout = imft_driver_info.eof_reply_timeout;
B483                                             /* the remote system replies only after
    backup_load_ has
B484                                                loaded everything, which can take a
    long time */
Preceding:
A482
A483                 call iox_$control (data_iocb_ptr, "read_reply_record", addr
    (local_icri), code);

Comparison finished: 1 difference, 3 lines.
```

The following `compare_ascii` output of the original and new `_imft_driver_info.incl.pl1` shows the proposed changes:

```
Inserted in B:
B45           2 eof_reply_timeout fixed bin (35),         /* seconds to wait for the reply
    to an EOF command; 0 => no limit */
Preceding:
A45           2 io_info,                                  /* data required to connect to
    the remote system */

Comparison finished: 1 difference, 1 line.
```

The following `compare_ascii` output of the original and new `cy73.errata.info` shows the proposed changes:

```
A1     :Info: cy73.errata: errata.cy73: IMFT_Reference: 2019-06-01 CY73 Errata
Changed by B to:
```

```
B1     :Info: cy73.errata: errata.cy73: IMFT_Reference: 2026-11-10 CY73 Errata


Inserted in B:
B65    Information for MR12.9:
B66
B67     pg 2-18: The values of the "version=" keyword are not "old" and
B68              "new"; the driver rejects both. CHANGE its description TO:
B69
B70              version= <2.0/3.0/4.0>
B71                Specifies the version of IMFT run by the remote system.
B72                "4.0" is the current version and the default. "2.0" is
B73                the version provided in MR10.1; with it, io_description
B74                and "allow_remote_request= yes" cannot be used.
B75
B76
B77     pg 2-18: After the "version=" keyword, ADD:
B78
B79              function= file_transfer
B80                Specifies the function of the driver. "file_transfer"
B81                is the only value and the default.
B82
B83              validate_system_id= <yes/no>
B84                Specifies whether the drivers validate each other with
B85                the passwords of the local_system and foreign_system
B86                Person_ids. If "no", no passwords are looked up, sent
B87                or checked, and it must also be "no" on the remote
B88                system's driver.
B89                This keyword is optional and defaults to "yes".
B90
B91
B92              min_time_est_msg= <N>
B93                Specifies, in seconds, how long a transfer must be
B94                expected to take for the output driver to log its
B95                estimated time. N must be from 30 to 3600. This keyword
B96                is optional and defaults to 60.
B97
B98              eof_reply_timeout= <N>
B99                Specifies, in minutes, how long the output driver waits
B100               for the remote system to reply to the end of a file or
B101               subtree. The remote system replies only after it has
B102               loaded all of the data, which can take longer than the
B103               transfer itself over a slow line. N must be from 0 to
B104               1440; 0 means no limit. This keyword is optional and
B105               defaults to 5. It cannot be specified for an input
B106               driver.
B107
B108
B109    pg 2-18: After the keywords above, ADD the following keywords, which
B110             make a connection through the answering service's dial
B111             facility instead of a fixed line:
B112
```

```
B113            in_dial= <qualifier>
B114              Specifies that the driver waits for a dial-in connection
B115              with this dial qualifier (see dial_manager_) and uses
B116              the channel that dials in. io_description,
B117              input_description and output_description are not used.
B118
B119            out_dial= <text>
B120              Specifies that the driver connects to the remote system
B121              by attaching its line as usual (normally a dial-out
B122              channel given by io_description), waiting for
B123              trigger_text, and then sending <text>, normally a dial
B124              preaccess request naming the remote driver's in_dial
B125              qualifier. The reply must contain " dialed to ".
B126
B127
B128            trigger_text= <text>
B129              Specifies the text the driver waits for before sending
B130              the out_dial text. It is required with out_dial.
B131
B132            sleep_time= <N>
B133              Specifies, in minutes, how long an automatic driver waits
B134              before retrying after an error in the dial connection
B135              sequence. N must be from 1 to 60. This keyword is
B136              optional and defaults to 5. It is used only with in_dial
B137              or out_dial.
B138
B139            debug_connect= <yes/no>
B140              Specifies whether the driver logs each line it sends and
B141              receives while making a dial connection. This keyword is
B142              optional and defaults to "no". It is used only with
B143              in_dial or out_dial.
B144
B145
B146            idle_line_drop= <yes/no>
B147              Specifies whether an output driver hangs up its dial
B148              connection when it has no requests, and dials again when
B149              a request arrives. This keyword is optional and defaults
B150              to "no". It is used only with in_dial or out_dial and
B151              cannot be specified for an input driver.
B152
B153            idle_delay_count= <N>
B154              Specifies how many times the output driver must be found
B155              idle before "idle_line_drop= yes" hangs up the line. N
B156              must be from 0 to 1000. This keyword is optional and
B157              defaults to 2.
B158
B159
B160    pg 2-18: After the keywords above, ADD:
B161
B162            debug_mode= <yes/no>
B163            copy_data= <yes/no>
B164              Are recognized only when the driver runs in test mode
```

```
B165              (iod_tables test entry). They attach imft_io_ with
B166              -debug and -copy_data. Both default to "no".
B167
B168
Preceding:
A65    :Internal: history_comment.gi: 2019-06-01 history_comment

Comparison finished: 2 differences, 106 lines.
```

3 Documentation

The `args:` keywords of the IMFT driver are documented in CY73-01, *Inter-Multics File Transfer Facility Reference Manual*, pages 2-16 to 2-18, but that list is out of date: it gives `version=` the values `old` and `new`, which the driver rejects (it takes `2.0`, `3.0` and `4.0`), and it leaves out 12 keywords the driver accepts: `function=`, `validate_system_id=`, `min_time_est_msg=`, the dial-connection keywords `in_dial=`, `out_dial=`, `trigger_text=`, `sleep_time=`, `debug_connect=`, `idle_line_drop=`, `idle_delay_count=`, and the test-mode keywords `debug_mode=` and `copy_data=`.

This MCR adds an "Information for MR12.9" section to `>doc>info>cy73.errata.info` that corrects `version=` and documents all of these keywords, together with the new one:

```
        eof_reply_timeout= <N>
          Specifies, in minutes, how long the output driver waits
          for the remote system to reply to the end of a file or
          subtree. The remote system replies only after it has
          loaded all of the data, which can take longer than the
          transfer itself over a slow line. N must be from 0 to
          1440; 0 means no limit. This keyword is optional and
          defaults to 5. It cannot be specified for an input
          driver.
```

The descriptions are taken from the driver's code (`imft_driver_`'s initialization and its dial connection sequence).

4 Test

1. Without the keyword: an output driver initializes, and a large transfer that needs more than 5 minutes for the EOF reply still times out after 5 minutes, as before.
2. `eof_reply_timeout=30` on the output driver: the same transfer completes and the user is notified of the arrival.
3. `eof_reply_timeout=0`: the same transfer completes.
4. `eof_reply_timeout=1`: a transfer whose EOF reply takes more than a minute times out after 1 minute (checks that the value is used). A large subtree over a slow line (or to a heavily loaded receiver) gives a slow reply.
5. `eof_reply_timeout=1441`, `=-1`, `=abc`: initialization stops with the message above.

6. `eof_reply_timeout=5` on an input driver: initialization stops with "The "eof_reply_timeout" keyword may not be specified for an input driver."
7. Ordinary transfers in both directions, a remote request (`enter_imft_request -source`), and an aborted transfer behave as before.

5 Files Changed

`bound_imft_.s.archive`	`imft_driver_.pl1`
`bound_imft_.s.archive`	`imft_transmit_object_.pl1`
`bound_imft_.s.archive`	`imft_receive_object_.pl1` (recompiled only)
`>ldd>include`	`_imft_driver_info.incl.pl1`
`>doc>info`	`cy73.errata.info`

6 Version History

2026-11-10	1.0	Initial version of the MCR.