

To: MTB Distribution
From: Gary C. Dixon
Date: September 1, 1981
Subject: Deferred Trouble Reports

THE PROBLEM:

Many people have asked for some facility to move problem TRs to a state or priority which would not count against goaling statistics. Usually, these TRs report problems we have decided not to fix (because the problem has minor effects, or is in code we plan to replace, or because there are no resources to maintain the code, etc).

THE SOLUTION:

After discussing this with several individual contributors and with management, the best solution for this problem is to define a new deferred state which would resolve the TR. Several alternatives are available for the canned answer.

CANNED ANSWER Alternatives

- I. Work on this problem has been deferred.
- II. Work on this problem has been deferred until resources are available.
- III. Because this problem has minimal impact on system usage, correction has been deferred in favor of higher-priority work.

Such TRs would remain routed to the developer responsible for the system area associated with the problem, and would appear on that person's TR summary. However, the TR would be resolved and would no longer count against statistics.

Resolved TRs in the deferred state would still be easy to find in the TR system (should we someday decide to fix such problems) using `tr_query` selection. It would be easy to use `tr_query` to gather statistics on the number of deferred TRs at any given time. This would give management some idea of the impact which TR deferral is having on the product.

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

Note that deferred TRs could still appear in an error list associated with the related system area. An error list entry would not be mandatory, but could be useful in helping the developer track such deferred TRs. The interface between error list entry and TR would be implemented by defining a new `edit_error_list` status value, called `deferred`.

ALTERNATE SOLUTIONS:

Several people suggested other methods of handling such TRs. One proposal called for shifting such TRs to a lower, ungoaled priority. However, it would be difficult to get upper management to approve an ungoaled priority.

Another idea involved marking such TRs as limitations, documented only in an error list. However, this would require maintaining lists of these errors in an error list. Maintaining the error lists would mean more work for developers. Also, long lists of inconsequential errors might make Multics products appear buggier to customers.

The solution above was chosen because it avoids these problems.

DETAILED PROPOSAL:

If we define a deferred state, then we need to define criteria which specify when it is appropriate to use this state. I have reviewed about 1/2 of the external, normal problem reports and have come up with a possible criteria.

Criteria for Deferring TRs

1. The TR must already be a normal priority problem, AND
2. The problem described must have a minimal impact on individual users and on the user community as a whole. Typical areas of minimal impact are:
 - A) Problems in commands/subsystems we intend to replace in the future (ie, in software whose support level is declining). For example, we stopped fixing indent bugs when `format_pll` was in the wings.
 - B) Bugs which have no adverse affects. Example: `hcs_$star_` accepts dir names beginning with `>>`.
 - C) Problems which have an obvious bypass which is easy to use. Examples: `install` command requires `w` access to CDT being installed; looping when a memo invokes the memo command, etc.
 - D) Problems which are unlikely to occur. Examples: `gedx` leaves input mode when user types `\034`; help malfunctions on info seg containing only blank lines.
 - E) Problems which can be ignored by the user. Examples: misspelled words in documentation; misspelled words in command output or error messages; extra data (beyond what command is specified to return) in printed output; etc.

Of the TRs I surveyed, about 1/2 could be resolved with the deferred state by using the above criteria.

Note that, based upon the criteria above, the availability or lack of resources to fix a particular problem does not have direct bearing on the use of the deferred state. The criteria for deferral center on the problem having minimal impact on users.

QUESTIONS:

Which of the alternative canned answers (eg, I, II or III) would you prefer?

Can you propose additional instances of item (2) above for deferring problems because of their minimal impact? If we can clearly spell out the exact criteria, there will be fewer disagreements over deferrals.

Should some other major criteria instead of (or in addition to) items (1) and (2) above be used to control deferral?

How do you feel about the overall idea?

Comments and opinions may be forwarded via:

Continuum: TR_System (tr) continuum meeting on System M

Mail: GDixon.TR on System M

Phone: Gary Dixon at (HVN 341-7295)