INTRODUCTION

This MTB describes a command, edit_error_list, which allows the creation and maintenance of Multics Standard Error Lists, and provides an automated interface to the Multics Trouble Reporting System (MTRS). This command operates on a "lister" data base, allowing the entire spectrum of lister commands to be utilized to provide customized output.

PURPOSE

The use of the edit_error_list command by developers obviates the need for them to submit MTRS transactions separately from error list maintenance and provides for some degree of standardization of error lists. This standardization makes the automated submission of MTRS transactions possible without developer intervention, and allows multiple MTRS transactions for a single error list entry, when appropriate. Additionally, it will provide a simple (hopefully), and convenient method of error list maintenance for those developers who do not wish to create their own tools.

WHY LISTER?

Considerable comment, in the continuum error lists meeting, was generated about the choice of lister vs. MRDS/MRPG. Lister was chosen because it requires little work in the initial implementation and it (lister) already has the necessary capabilities to generate reports (or info secs) via a very simple mechanism. The major drawback of using a lister data base (as was mentioned in the error lists meeting) is the 255 record limitation. This limitation is currently being addressed by the lister developer, and hopefully will be resolved before the limit is reached by an error list.
CONVERSION

Initially, I don't believe I can provide a general enough tool to convert existing bug lists. However, after the initial implementation of edit_error_list, I will address a method of conversion by those who want it. Suggestions as to methods of conversion would be entertained, but implementation time must be a prime consideration in any proposal.

TR SYSTEM INTERFACE:

An interface will be provided between existing error list maintenance tools and the TR system, in the form of an extended answer_trouble_report command.

A new -error_list_info (-eli) control argument has been added. Its usage is:

-eli error_file_name error_file_entry_number

For example,

-eli pl1 2537

This new field references an error list entry associated with the TK. The error list information will appear as the answer to the "error list info" field when the TR is printed.

A new -linked_tr (-ltr) control argument has been added. Its usage is:

-ltr {numbers}

where the numbers which follow are TK numbers of reports associated with the report being answered. These associated reports will inherit the answer given in this invocation of atr. This avoids invoking atr for each of the reports. The associated TK numbers will appear as the answer to the "linked trs" field.

The operands for the -linked_trs control argument are optional to facilitate use of -linked_trs from error_list maintenance tools. For example, if an error list entry contains 5 associated TK numbers, the error list tools which changes the state of the entry could invoke atr as follows:

do "atr -rql -rq send &4 -state &3 -eli &1 &2 -linked_trs &f5"

with arguments which were:

error_list_name error_list_entry_number new_state assoc_tr_numbers

Exactly the same calling sequence would work if only one associated TK number were present in the error list entry.
Name: edit_error_list

The edit_error_list command performs the functions necessary to create and maintain a Multics "Standard" Error List. In the default mode of operation, edit_error_list enters a request loop where it reads user requests from the terminal until the quit request causes it to exit.

For the sake of simplicity and flexibility, edit_error_list operates on listin files (See AZ98 Section 7 for the description of lister and listin files). In order to implement certain requests, the comment feature of lister will be employed. Specifically, the next_entry_number, set_default_value, set_prompt (on off) requests will be implemented to store certain values in lister comments. It is imperative for the proper operation of edit_error_list that these comments not be moved, changed or deleted except via the appropriate edit_error_list request.

The "interface" employed by edit_error_list with the MTAS is via the answer_trouble_report command.

Usage:
edit_error_list NAME {-control_args}

where:

1. NAME

is an error list name. If NAME does not have a suffix of error_list, then one is assumed. However, the suffix error_list must be the last component of the segment. If NAME contains a "<" or a ">" then it will be used as the pathname of the error_list. Otherwise, the error_list search list will be used to find NAME. The default error_list search list is >doc>iis>error_lists.

2. control_args can be chosen from the following:

-request STR, -rq STR
specifies an initial request line to be executed prior to entering the request loop.

-request_loop, -rql
specifies that the edit_error_list request loop be entered. This is the default.

-quit
exits after performing any operations specified by control arguments. The default is to enter the
request loop.

-prompt STR
changes the prompt for edit_error_list request lines to STR. If STR is "", the user is not prompted.

-no_prompt
stops prompting for edit_error_list requests. The default prompt is "edit_error_list(N):", where N is the recursion level if greater than one. Equivalent to -prompt "".

REQUEST LINE SYNTAX:

A edit_error_list request line beginning with a ""."" causes the rest of the line to be passed to the command processor.

The semicolon is used to separate multiple requests on a line.

There are thirty-three requests to edit_error_list, they are;

create, add, select,
change, set output, set prompt on,
set prompt off, set default value,
add_field, print,
edit, qedx, ted, teco, emacs,
copy, move, delete, atr,
set atr on, set atr off, set atr args,
fill, apply, enter,
abort, execute,
help, set next entry,
"?", "".
update and quit.

Each edit_error_list request accepts a particular set of arguments, as described under "REQUESTS" below. In general, requests operate on an "entry". An entry is a complete lister record. Since these "Standard" error lists are in reality lister listin files, any lister command may be used by the developer to produce customized output. It must be noted, however, that the edit_error_list command is the only way to have changes to the error list reflected in the "status.info" file for the particular area, and have Trouble Report transactions submitted based upon error list changes without additional intervention.
REQUESTS:

create {-control_args}, cr {-control_args}
causes the creation and initialization of an error list.
This "initialization" is actually the construction of a
lister file "header" and the setting of certain per error list
values.

Acceptable control_args to the create request are:

-record_delimiter CHAR, -rd CHAR
sets the error list record delimiter to CHAR. CHAR may be
chosen from the following set: ! # & $ * = ? @ ^ ! ~
The default record delimiter is @.

-field_delimiter CHAR, -fd CHAR
sets the error list field delimiter to CHAR. The
record delimiter and field delimiter may not be the
same CHAR. See above for acceptable values for CHAR.
The default field delimiter is #.

-field_name f1 d1 f1 d2 ... f1 dn, -fn f1 d1 f1 d2 ... f1 dn
allows specification of error list fields in addition
to the "standard" predefined fields (see Standard
Error List Field Descriptions, below). A fieldname
may not begin with a "-" character. Prompting occurs
for these fields by default, see the -prompt_off control
argument below.

-next_entry_number N
initializes the internal error list "next" entry
number to N (1 < N < 99999999). This is the number
that will be assigned to the entry in the next add
request. The default is 1.

-pathname pathname, -pn pathname
specifies the pathname of the status.info segment to
be updated when the update request is invoked.
Pathname may be an absolute or relative path. The
default pathname is >doc>iis>NAME.status.info.

-prompt_off f1 d1 f1 d2 ... f1 dn
"turns off" prompting for f1 d1. This control argument
may be used for all fields defined by the -field_name
control argument and the "standard" fields.
-set_default fld1 value1 fld2 value2...fldn valuen
sets the default value for fldi to valuei. This control argument may be used to set the default value for any fields defined by the -field_name control argument and all "standard" fields except numb.

-atr_args "STR"
sets the arguments supplied to the answer_trouble_report command to "STR". STR is passed to atr when MTRS transactions are submitted. No validation of these arguments is provided by eel.

-auto_atr_off
turns "off" the automatic submission of MTRS transactions. The default is to automatically submit MTRS transactions when appropriate.

add {-control_args}
adds an entry to the error list. The entry number is generated automatically from the next_entry_number field.

Acceptable control arguments to the add request are:

-entered_date DATE, -edt DATE
sets the entered_date field to DATE (actually yyyy-mm-dd). DATE must be acceptable to the convert_date_to_binary_subroutine. The default is the current date.

-status VALUE, -st VALUE
sets the status field to VALUE (See Standard Error List Field Descriptions (Table 1), below for valid status values and their meanings). The standard default is "error".

-status date DATE, -stdt DATE
Sets the status date field to DATE (actually yyyy-mm-dd). DATE must be acceptable to the convert_date_to_binary_subroutine. The default is the current date.

-tr trid1 trid2...tridn
sets the tr field to "canonicalized" tr identifiers, i.e. phxnnnnnn for all TR ids with the "TR" or "phx" prefix, and trids that are all numeric. Changes of this entry's status will automatically generate appropriate TR system transaction(s) for all phxnnnnnn numbers, all others are ignored.
**-type STR**
sets the type field to STR. Valid values for type are "problem", "suggestion", and "limitation". The standard default is "problem".

**-desc STR**
sets the desc field to STR.

**-bypass STR**
sets the bypass field to STR.

**-version STR**
sets the version field to STR.

**-added_info STR, -ai STR**
supplies the "added_info" field, required by some changes of status, to the generated TR system transaction(s).

**-field_name fieldname STR, -fn fieldname STR**
sets field "fieldname" to STR. If "fieldname" is not defined, an error occurs. This control argument allows setting of error list fields in addition to the "standard" predefined fields (See Standard Error List Field Descriptions, below). This control argument may appear more than once.

**select "STR", sel "STR", sl "STR"**
allows selection of entries based upon the content of one or more fields. The select request, and active request, simply prints or returns the entry numbers meeting the selection criteria. The STR argument consists of one or more field comparisons. A field comparison involves comparing a test string to the value of the specified field in current record. The field comparison statement always consists of three parts:

"fieldname comparison_operator test_string"

*fieldname* is the name of a field defined in the error list.

*comparison_operator* specifies what comparison is performed. The opposite comparison is performed if the comparison operator is preceded by "not". The comparison operators are:

contains test string is contained in the field value. The comparison is made without regard to case, i.e. uppercase letters
**edit_error_list**

*compare equal to lowercase letters.*

**equal**  
Test string is equal to the field value. Uppercase letters are distinct from lowercase letters with this operator.

**greater**  
Field value is alphabetically greater than the test string (e.g., 0123456789Aa ... Zz).

**less**  
Field value is alphabetically less than the test string.

**test_string**  
is the string that is compared to the field value string.

Several field comparisons may be specified by the STR argument. Field comparisons are combined by the logical operators "and," "or," or "not." In the absence of parentheses, the prefix "not" operator is evaluated first, then the infix "and" operator, then the infix "or" operator. Parentheses may be used to specify the exact order of evaluation. These rules are similar to the PL/I rules for Boolean expressions.

**change {N} {S} {-control_args}, c {N} {S} {-control_args}**  
changes the status field of entry number N to S (see Standard Error List Field Description, below for valid status codes). If N is not supplied, the current entry is changed. If S is not supplied, prompting occurs for "promptable" fields not supplied by appropriate control arguments. This request sets the "current" entry to N.

Acceptable control_args to the change request are:

- **-status date DATE, -stdt DATE**  
Sets the status date field to DATE (actually yyyy-mm-dd). DATE must be acceptable to the convert_date_to_binary subroutine. The default is the current date.

- **-tr trid1 trid2...tridn**  
Sets the tr field to "canonicalized" tr identifiers, i.e. phnxxxxx for all TR ids with the "TR" or "phx" prefix and trids that are all numeric. Changes of this entries status will automatically generate appropriate TR system transaction(s) for all phnxxxxx numbers, all others are ignored.

- **-type STR**  
Sets the type field to STR. Valid values for type are
edit_error_list

"problem", "suggestion", and "limitation". The standard default is "problem".

_desc STR
   sets the _desc field to STR.

_bypass STR
   sets the bypass field to STR.

_version STR
   sets the version field to STR.

_added_info STR, -ai STR
   supplies the "added_info" field, required by some changes of status, to the generated TR system transaction(s).

_field_name fieldname STR, -fn fieldname STR
   changes field "fieldname" to STR. If "fieldname" is not defined, an error occurs. This control argument allows changing of previously defined error list fields in addition to the "standard" predefined fields (See Standard Error List Field Descriptions, below). This control argument may appear more than once.

_enter
   causes the change to be "entered" in the error list.

set_output {-control_args}, so {-control_args}
   specifies the format and content of the print request. If control_args are omitted then -format long is used.

Acceptable control_args to the set_output request are:

_format NAME
   uses NAME.listform to control the format and content of the output. If NAME does not have a suffix of listform, then one is assumed. However, the suffix listform must be the last component of the segment. If NAME contains a "<" or a ">" then it will be used as the pathname of the listform. Otherwise, the error_list search list will be used to find NAME. The default error_list search list is >doc>iis>error_lists.

-field_name fld1 fld2... fln, -fn fld1 fld2...flrn
   prints only the specified fields, one to a line. Fldi must be a defined fieldname in the error list.
edit_error_list

add field fieldname1 fieldname2...fieldnamen
  defines fieldname1 as a valid field in the error list.
  Prompting occurs for these fields by default. (see
  "Prompting" below).

set_prompt_on fld1 fld2...fldn
  causes fldi to be prompted for if its value is not
  supplied via a control argument. Fldi must have been
  previously "defined".

set_prompt_off fld1 fld2...fldn
  Stops prompting for fldi. Fldi must have been previously
  "defined" (see "Prompting" below).

set_default_value fld1 value1 fld2 value2...fldn valuen
  Sets the default value for fldi to valuei. This request
  may be used to set the default value for any fields
  "defined" by the -field_name control argument and all
  "standard" fields except numb.

print {N} [-control_args], pr {N} [-control_args]
  prints the current (default) entry, or entry N if specified.
  If -control_args are omitted, then the current "set_output"
  specification is used to control the output.
  This request sets the "current" entry to N.

Acceptable control_args to the print request are:

  -format NAME
    uses NAME.listform to control the format and content of
    the output. If NAME does not have a suffix of
    listform, then one is assumed. However, the suffix
    listform must be the last component of the segment. If
    NAME contains a "<" or a ">", then it will be used as the
    pathname of the listform. Otherwise, the error_list
    search list will be used to find NAME. The default
    error_list search list is >doc>iis>error_lists.

  -field_name fld1 fld2... fln, -fn fld1 fld2...fldn
    prints only the specified fields, one to a line. Fldi
    must be a defined field_name in the error list.
allows the user to edit an error list entry. If N is specified, then entry N is edited. The default is the current entry. This request sets the "current" entry to N.

apply {N} COMMAND-LINE, ap {N} COMMAND-LINE
execute a Multics command line for the specified entry. The command line is modified to include the pathname of a segment containing the entry. If N is omitted, the current entry is used.

enter {-control_args}
The enter request is used to update the NAME.error list with the additions/changes made thus far. If necessary, generates appropriate TR system transaction(s), and optionally exits the edit_error_list request loop.

Acceptable control_args to the enter request are:
-quit
exits the edit_error_list request loop after all validation and prompting are completed. The default is to return to the edit_error_list request loop.

copy {N} NAME, cp {N} NAME
copies entry N to NAME.error_list. If N is not supplied, then the current entry is moved. NAME is an error list name. If NAME does not have a suffix of error_list, then one is assumed. However, the suffix error_list must be the last component of the segment. If NAME contains a "<" or a "">
then it will be used as the pathname of the error_list. Otherwise, the error_list search list will be used to find NAME. The default error_list search list is >doc>iis>error_lists.

move {N} NAME, mv {N} NAME
copies, then deletes, entry N to NAME.error_list. If N is not supplied, then the current entry is moved. NAME is an error list name. If NAME does not have a suffix of error_list, then one is assumed. However, the suffix
error_list must be the last component of the segment. If
NAME contains a "<" or a ">", then it will be used as the
pathname of the error_list. Otherwise, the error_list
search list will be used to find NAME. The default
error_list search list is >doc>ii>error_lists.

delete {N}, dl {N}
deletes entry N from the error list. If N is not supplied,
then the current entry is deleted.

atr {N} {-control_args}
invokes the answer_trouble_report command for entry number
N. If N is omitted, then the current entry is used. This
request uses the the default atr control arguments specified
by the set_atr_args request or specified by the -atr_args
control argument in the create request.

Acceptable control_args to the atr request are:

-argument "STR", -aq "STR"
"STR" are control arguments for the answer_trouble_report
command. STR is appended to the default atr control
arguments (if any) before the atr command is invoked.
No argument validation is provided by eel.

set_atr_on
Turns "on" the automatic submission of MTRS transactions.
This is the default.

set_atr_off
Turns "off" the automatic submission of MTRS transactions.

set_atr_args "STR"
Sets the arguments given to the answer_trouble_report
command to "STR". No validation of these arguments is
performed by eel.

fill {N} {-control_args}
causes filling of the multi-line fields of entry number N.
If N is omitted, then the current entry is filled.

Acceptable control_args to the fill request are:

-line_length n, -ll n
re-formats the entry according to the filling
algorithm of format document to column n. The default fill column is 72.

abort [-control_args]
"aborts" the current entry without validation and related activities. The current entry is not added to or modified in the error list.

Acceptable control_args to the abort request are:

-quit
exits the edit_error_list request loop. The default is to return to the edit_error_list request loop.

-force, -fc
forces the exiting of the request loop without query if an entry has been modified without being "entered".

execute STR, e STR
passes the concatenation of STRs with intervening spaces to the Multics command processor. This request is different from "." because it is first parsed and interpreted as a edit_error_list request line before being passed to the command processor.

help {request_name}
provides information on the various edit_error_list requests. If request_name is given, then information on that request is provided. If no request_name is given, information on edit_error_list is provided.

set next_entry N
allows the modification of the "next_entry_number", that number given to the next add request, to N (1 ≤ N ≤ 99999999).

? lists the available edit_error_list requests and request line syntax.

"." edit_error_list identification request.
update {path}
    updates path with the contents of NAME.error_list using
    the current error_list.listform. If path is not specified
    the value specified, or assumed, by the -pathname control
    argument in the create request is used.

quit {control_args}
    exits the edit_error_list request loop.

Acceptable control_args to the quit request are:

    -force, -fc
    forces the exit of the request loop without query if an
    entry has been modified without being "entered".

PROMPTING:

    The default is to prompt for all defined fields. This
    default may be overridden by the set_prompt_off request, or by
    the use of the -prompt_off control argument in the create
    request. Note that the "added info" required by the MTRS in
    some instances is not part of the error list and will be
    prompted for when required unless the -added info control
    argument is used to provide it. A "." on a line by itself
    terminates the input for multi-line fields.
<table>
<thead>
<tr>
<th>Field Name</th>
<th>R → required</th>
<th>Description</th>
<th>Validation criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>numb</td>
<td>R</td>
<td>the entry number (unique to this list) of this error.</td>
<td>(8) 0-9</td>
</tr>
<tr>
<td>entered_date</td>
<td>R</td>
<td>the date (yyyy-mm-dd) this entry was entered</td>
<td>cdtb</td>
</tr>
<tr>
<td>status_date</td>
<td>R</td>
<td>the date (yyyy-mm-dd) of this entries CURRENT status.</td>
<td>cdtb</td>
</tr>
<tr>
<td>type</td>
<td>R</td>
<td>type of this entry ex. problem or suggestion</td>
<td>problem suggestion limitation</td>
</tr>
<tr>
<td>status</td>
<td>R</td>
<td>the CURRENT status of this entry</td>
<td>error long name short name error er exl problem dev limitation dev exl submitted dev dev limitation submitted li accepted dev submitted sub accepted accepted a rejected rejected r entered entered e</td>
</tr>
<tr>
<td>tr</td>
<td>O</td>
<td>one (or more) TR identifiers that describe this error. The TR system 'links' 'known' associated TR's.</td>
<td></td>
</tr>
<tr>
<td>desc</td>
<td>R</td>
<td>a description of the error or a scenario that will reliably reproduce the error.</td>
<td>ASCII text</td>
</tr>
<tr>
<td>bypass</td>
<td>O</td>
<td>a description of how to fix or bypass the error.</td>
<td>ASCII text</td>
</tr>
<tr>
<td>version</td>
<td>O</td>
<td>the version of the software that include the fix for the problem.</td>
<td></td>
</tr>
</tbody>
</table>

Table 1.
Standard Error List Field Descriptions