

To: MTB Distribution  
From: Rickie E. Brinegar  
Date: May 26, 1981  
Subject: LINUS changes for MR9.0

## I. INTRODUCTION

This MTB describes the user visible changes in the Logical Inquiry and Update System (LINUS) for the MR9 software release. This will be done in two sections. The first section will cover the changes from MR6.5 to MR8.0. The second section will cover changes from MR8.0 to MR9.0.

## II. MR6.5 TO MR8.0 CHANGES

### A. SYNTAX CHANGES

The following control arguments were added to the linus command:

- no\_prompt  
This permits the user to turn off the prompting of the lila editor and the LINUS subsystem at the time the subsystem is entered from command level.
- set\_lila\_prompt\_string STR, -slaps STR  
This permits the user to define what the prompt string for the lila editor will be at the time that the LINUS subsystem is entered from command level.
- set\_linus\_prompt\_string STR, -slups STR  
This permits the user to define what the prompt string for the LINUS subsystem will be at the time that the subsystem is entered from command level.

The existing . request was documented for both the LINUS subsystem and the LILA editor.

The following new request were implemented:

- .. The text following the ".." is passed directly to

---

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

the Multics command processor. This was implemented for both the LINUS subsystem and the LILA editor.

assign\_values VARIABLE\_LIST, av VARIABLE\_LIST

This permits the user to assign values to the variable name in the VARIABLE\_LIST until there are no more variables in the VARIABLE\_LIST or there is no more data to be retrieved from the database for the current selection expression. If the data is used up before the VARIABLE\_LIST, new variables in the variable list will not be created and old variables will retain their old values.

This functionality is the same as that which had previously been described for the set request. The implementation of the assign\_values request was done to eliminate the confusion between the function that the assign\_values request (previously the set request, which is now an obsolete request but still works) performs and the set operations that are possible in the lila selection expression.

delete\_temp\_table STR, dltn STR

This allows the user to throw away the temporary table whose name appears as STR.

list\_values {VARIABLE\_LIST}, lv {VARIABLE\_LIST}

This permits the user to look at the values of the LINUS variables that have been assigned values using either the assign\_values or the set requests.

The modify request was changed to accept the "" string for the cases of database attributes that are declared as character, character varying or bit varying data types.

The print request had the following control arguments added:

-col\_width\_trunc WIDTH\_LIST, -cwt WIDTH\_LIST

This allows the user to specify that, instead of printing asterisks in the columns where the widths were too small for the value, the values are to be truncated without any notification of the truncation.

-no\_end

This allows the user to specify that the "(END)" string is not to be printed upon completion of printing of the desired data.

The set\_mode request had the following modes added:

set\_lila\_prompt\_string STR, slaps STR

This permits the user to specify the prompt string to be used while in the LILA editor. The default string is "->".

set\_linus\_prompt\_string STR, slups STR

This permits the user to specify the prompt string to be used while in the LINUS subsystem, but outside of the LILA editor.

### C. ADDITIONAL DOCUMENTATION CHANGES

The modify request had two restrictions that were documented in the MR8.0 manual. They are:

1. Expressions that are to be used as new column values when doing a modify must be enclosed in parenthesis.
2. Any literal value that is to be used as a new column values which contains the blank character must be enclosed in quotes.

## III. MR8.0 TO MR9.0 CHANGES

### A. DEFINITIONS

For definitions of secure database and database administrator (DBA), see [1].

For the purposes of this MTB, secure data submodels are MRDS submodels that reside under the database in the secure.submodels directory. See [1] for a more detailed discussion of accessing a secure database.

### B. SYNTAX CHANGES

For compatibility with other Multics commands the following changes have been made to LINUS:

The short name dl was added to the delete request. The former short name, d, was undocumented.

The short name pr was added to the print request. The former short name, p, was undocumented.

The short name `cls` added to the `create_list` request. The former short name, `cl`, was undocumented.

The control argument `-file`, with short name `-f`, for `store` was changed to `-input_file`, with a short name of `-if`. The `-file` and `-f` were retained as acceptable forms of the control argument, but were undocumented.

In addition the `set` request, which was marked as obsolete in the MR8.0 release, was undocumented.

A new operation code was added for the `set_scope` and `delete_scope` requests. The code `"u"` was added for updated operations. When `"u"` is encountered, it is treated as the string `"smd"`, and, in fact, the `list_scope` request will list `"smd"` as being the scope set if `"u"` was used to set scope.

#### C. ADDITIONAL DOCUMENTATION CHANGES

Some of the LINUS users who are more familiar with MRDS have expressed some confusion at the way that LINUS builds keys for temporary tables. The order that the columns will appear in the temporary table is defined by the selection expression that is in effect. Therefore the order of columns in the key are also defined by the order that they are found in the selection expression. The ordering of columns in the key will impact performance.

The performance question comes into being when one is aware of what columns from the table may be used as key heads by MRDS. Searching by key head is, in general, quicker than sequentially searching data. As `define_temp_table` (`dtl`) was previously documented, one might infer, although it was not stated, that the order that the column names were given in the `dtl` request would be the order that they would appear in the key, thereby, expecting certain queries to run faster than they did if the column names were not correct in the selection expression.

The `set_scope` request that asks for a modify, store or delete `permit_op` will no longer automatically add a request for a `retrieve permit_op`. This change was made as a result of changes in the MRDS scope mechanism.

#### D. EFFECTS OF SECURITY

When LINUS is used to interface to databases or data submodels which are not currently in a secured state, the user will not notice any change in behavior in LINUS from the MR8 release, other than the bug fixes which have been made.

The most notable changes in user interfaces because of security are in the open and list\_db requests. In both cases, the changes will only be seen by users who are not DBAs on secure databases. The changes that will be seen are:

open - The non-DBA user of a secure database will not be permitted to open a database via the data model, that is, by giving the pathname of the database. All openings for the non-DBA user must be through a secure data submodel.

list\_db - When doing a list\_db (ldb) on a secured database the non-DBA user will not have the domain names of the database displayed. In addition, ldb will not display what columns make up the key. The first column of the key will be displayed as an indexed column, while the other columns making up the key will be displayed as data columns.

The other change that has come about due to the security work involved in [3] is in the set\_scope request. Some of the inconsistencies between what scope the user is asking to have set and what scope may actually be set are now caught and reported at the time of execution of the set\_scope request. As a result, some error messages will be seen at set\_scope time where the user use to see incorrect access violations at retrieve/update time. The MRDS error codes reporting these inconsistencies will be trapped and converted to LINUS error codes which do not use the words tuple and attr (for attribute). This is being done because, throughout the documentation of LINUS, tables, rows and columns are used instead of relations, tuples and attributes, respectively, in order to make the use of LINUS more understandable to users who are not "into" relational databases.

#### IV. DIFFERENCES BETWEEN THIS MTB AND [2]

It was originally felt that the LINUS list\_scope, set\_scope and delete\_scope request should reflect the new terminology used by MRDS in regards to scope access settings. This would have meant using an "a" instead of an "s" as the LINUS code for the scope request for store operations, the "a"

standing for append, where MRDS uses append tuple, and the "s" standing for store, where LINUS has the store request.

The use of the "a" was deemed to be somewhat confusing for users who were not familiar with MRDS where the "s" has a more readily apparent derivation and is consistent with the other LINUS scope access codes. For this reason, LINUS will continue to use the "s" and the "a" will not be introduced.

Notice should be taken that this is merely a terminology difference between LINUS and MRDS in much the same vein as the terminology differences of relations and tables, tuples and rows, and attributes and columns. LINUS now depends on MRDS to keep track of what scope is currently set and no longer keeps this information in internal static storage.

#### V. REFERENCES

- [1] MTB-501, The New MRDS Security Approach, Jim Gray
- [2] MTB-502, Effects of Security on the MRDS Interface, Jim Gray
- [3] MTB-506, Extension to the create\_mrds\_dsm and display\_mrds\_dsm for MRDS Security, NS Davids