

To: MTB Distribution  
From: J.R.Ohlin  
Date: February 11, 1980  
Subject: The LINUS Test Facility

## INTRODUCTION

This MTB explains the LINUS test facility, where it is, what it does, and how to use it.

The Logical Inquiry and Update System (LINUS) is a versatile, easy to use, end user facility, which is used to access MRDS data bases, and to prepare data for report generation. LINUS may also be used to load, retrieve and access private data bases. Data to be acted upon is specified using the Linus Language (LILA), which is a simple high level language, designed to be used by non-technical users.

The LINUS test facility is located in the Multics System Test Library (>udd>STL>linus). All Multics project personel have status access to the library, but will require more access in order to run the LINUS tests. To request that access, please send a message to the STL Project Administrator (JOhlin.Multics). This facility tests pre-MR 8.0 LINUS.

The current test consists of a series of exec\_coms that exercise LINUS commands and builtins, places the results in an output segment which is then compared against a standard. The LINUS commands tested are create\_list, define\_temp\_table, execute, invoke, lila, list\_db, modify, open, print, quit, report, set, store, and write. Those not tested are close, declare, del\_scope, delete, help, list\_scope and set\_scope. One (intersection) of the three set operations (intersection, union and difference) is tested. Two (and &, or |) of the three logical operators (and &, or |, not ^) are tested. Three (=, ^=, >) or the six relational operators (>, <, <=, >=, =, ^=) are tested. Two (\*, +) of the four arithmetic operators (+, -, \*, /) are tested.

## TEST FLOW

The top level exec\_com (test\_linus.ec - See Attachment 1) first deletes the output file from the last execution, does a file\_output to that file and then calls the main test execom (linus\_dept\_store.test.ec). This main execom, using qx, inputs a MRDS data base model (dept\_store.cmdb), and then creates an unpopulated MRDS data base (dept\_store), containing five relations ( See Attachment 2 ). The execom then enters LINUS and

opens that data base in the exclusive update mode. The next step is to populate (store) the five tables from existing data files. A long series of LILA expressions are then generated to test most LINUS requests, arithmetic operators, logical operators, relational operators, and set operations as well as the LILA language itself.

The where clause builtin test (builtin.test.ec) is called next which tests all MRDS builtins with the exception of search and verify. (i.e. abs, after, before, ceil, concat, floor, index, mod, reverse, round and substr) Two of the five exclusively LINUS builtins, min and max are not tested in this test, but the other three, avg, count and sum are tested.

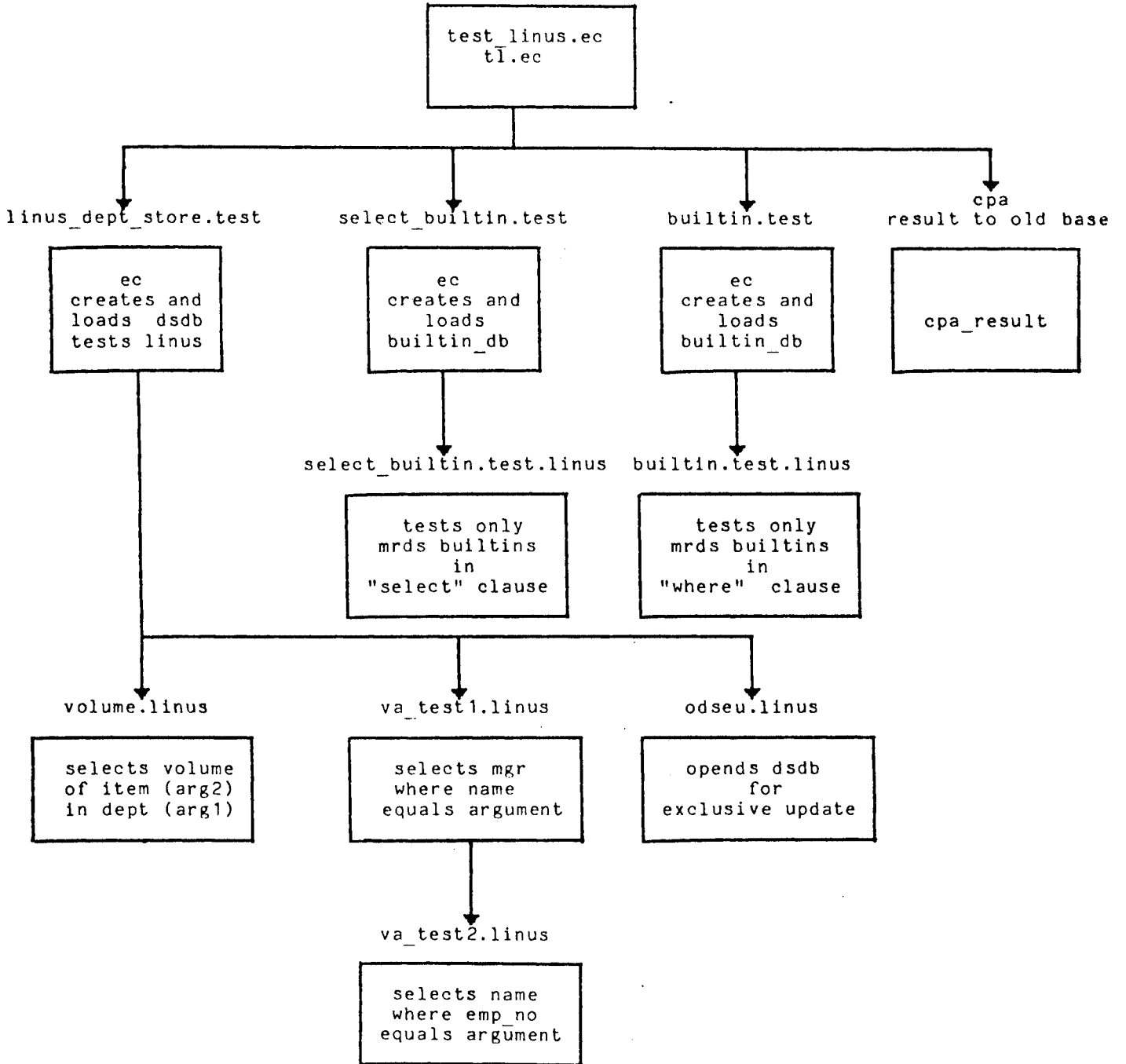
The select clause builtin test (select\_builtin.test.ec) is called as the final test and again as in the test above, all MRDS builtins with the exception of search and verify are tested in the select clause instead of the where clause.

Output is then returned to the console and the qx editor is entered to strip all trailing blanks from the output lines. This stripping is done to compensate for a pre-MR8.0 LINUS bug.

A compare\_ascii is then done comparing this output file with a standard output file. The compare\_ascii result is then printed on the console.

FLOW CHART

LINUS REGRESSION TESTS



USAGE

After receiving the proper access from the STL Project Administrator, to the LINUS test facility, one simply types those lines below that are preceded by an exclamation mark(!).

```
! cwd >udd>STL>linus
```

```
! ec test_linus
```

No data was found that satisfied the selection expression.

ERROR IN LINE 0020

A syntax error has been detected in a select clause.

```
count
```

```
count { select sales.item
```

```
r new_test_result
```

```
1,$s/-$$/
```

```
w
```

```
q
```

```
cpa_result
```

```
02/04/80 1406.7 mst Mon
```

Segments are identical.

```
r 14:06 176.298 1698
```

Note: The two error messages, as shown above, are emitted during the test.

ATTACHMENT 1

test\_linus.ec

```
&command_line off
dl new_test_result
fo new_test_result
ec linus_dept_store.test
ec builtin.test
ec select_builtin.test
co
&attach
qedx
r new_test_result
1,$s/*$//
w
q
&detach
dl cpa_result
fo cpa_result
cpa old_test_result new_test_result
co
pr cpa_result
&quit
```

ATTACHMENT 2

DATA MODEL FOR DATA BASE dept\_store  
 Created using dmd\_version 3  
 Created by JOhlin.STL.m  
 Created on 02/05/80 0236.7

RELATION NAME: class

Number attributes: 2  
 Key length (bits): 180  
 Data length (bits): 36

ATTRIBUTE	DECLARATION	DOMAIN	TYPE	OFFSET	LENGTH
item	character (20)	item	key	0	180
type	character (4)	type	data	0	36

RELATION NAME: emp

Number attributes: 6  
 Key length (bits): 270  
 Data length (bits): 396

ATTRIBUTE	DECLARATION	DOMAIN	TYPE	OFFSET	LENGTH
name	character (30)	name	key	0	270
emp_no	real fixed binary (17,0)	emp_no	data	0	18
dept	character (12)	dept	data	18	108
mgr	real fixed binary (17,0)	emp_no	data	126	18
sal	real fixed decimal (13,2)	sal	data	144	126
comm	real fixed decimal (13,2)	comm	data	270	126

ATTACHMENT 2 (Cont.)

RELATION NAME: loc

Number attributes: 2  
 Key length (bits): 108  
 Data length (bits): 9

ATTRIBUTE	DECLARATION	DOMAIN	TYPE	OFFSET	LENGTH
dept		dept	key		
	character (12)	unaligned	0	108	
floor		floor	data		
	real fixed binary (8,0)	unaligned	0	9	

RELATION NAME: sales

Number attributes: 3  
 Key length (bits): 288  
 Data length (bits): 36

ATTRIBUTE	DECLARATION	DOMAIN	TYPE	OFFSET	LENGTH
dept		dept	key		
	character (12)	unaligned	0	108	
item		item	key		
	character (20)	unaligned	108	180	
vol		vol	data		
	real fixed binary (35,0)	unaligned	0	36	

RELATION NAME: supply

Number attributes: 3  
 Key length (bits): 450  
 Data length (bits): 36

ATTRIBUTE	DECLARATION	DOMAIN	TYPE	OFFSET	LENGTH
supplier		supplier	key		
	character (30)	unaligned	0	270	
item		item	key		
	character (20)	unaligned	270	180	
vol		vol	data		
	real fixed binary (35,0)	unaligned	0	36	