Published: 10/30/69

Identification

Multics Segment List Data Insertion Command
msl_add
Edwin W. Meyer, Jr., Janice H. Cecil

Purpose

msl_add is a special purpose command intended to allow the addition of a limited set of entry types and items during the creation of a Multics Segment List.

Overview

There are three entries to msl_add, each specifying a different basic name type that may be added. Upon entry to the command, msl_add attempts to initiate the MSL segment specified in one of the arguments. If it can not be found, msl_add creates an empty MSL and prints the message "creating virgin msl". When it is ready to receive internal requests it types "go ahead" and listens for request lines. No further messages will be printed, so that request lines may be queued ahead without danger of I/O conflict.

The request line "." stores the MSL and exits the command. Other request lines with format peculiar to the specific entry enter data into the MSL. Each request line specifies the addition of data pertaining to a single name. If an entry for that name already exists in the MSL, the items to be added for that name are written over any existing items. Otherwise an entry for the name is created and the items added. Request line items are delimited by blank characters. The current date from a call to clock_ is used for the source and object installation dates.

$source

Request lines issued under this entry add source type (code 0-19) names to the MSL. The request format is:

```
name type who_auth source_archive document -bound_name
```

where type is the 1-2 letter abbreviation, and bound_name is an optional request argument. If any argument is null, it should be typed "*".

The following items are added to the MSL for the entry `name'.

Item no.	item id	description
0	name	from request line
1	type_code	from request line
2	source_instal	from call to clock_
3	object_instal	from call to clock_
5	who_auth	from request line
7	area_use	from request line
8	document	from request line
9	superior_list	"bound_name" added if present in request line
11	path_list	
11.0	source_path	source_dir (command)|| source_archive (request)
11.1	object_path	if "bound_name" absent from request, "object_path" from command. Otherwise null.
11.2	old_dir	from command

All other items remain null.

Usage

```
call msl_add$source (msl_path, source_path, object_path,
                     old_dir);
```

1) msl_path(char(*)) name of MSL to be edited (path or wdir entry name)

2) source_path(char(*)) MSL entry item

3) object_path(char(*)) MSL entry item

4) old_dir(char(*)) MSL entry item

$incl

Request lines issued under this entry add include type names to the MSL. The request format is:

incl_name area_use -incl_type-

Where incl_type is an optional argument (2 letter abbreviation) that specifies the include type of the entry. If the argument is absent, the type is the same as that of the previous request line. The initial default type is "ie" (incl.epl).

The following items are added to the MSL for the entry "incl_name":

item no.	item-id	description
0	name	"incl_name" from request
1	type_code	from request
2	source_instal	from call to clock_
3	object_instal	from call to clock_
7	area_use	from request line
11	path_list	
11.0	source_path	"incl_path" from command
11.2	old_dir	"incl_old" from command

All other items remain null.

Usage

call msl_add$incl (msl_path, incl_path, incl_old);

1) msl_path(char(*)) name of MSL to be edited

2) incl_path(char(*)) MSL entry item

3) incl_old(char(*)) MSL entry item

$bound

Request line issued under this entry add bound type names to the MSL. The request format is:

bound_name area_use

The following items are added to the MSL for the entry "bound_name":

item no.	item-id	description
0	name	"bound_name" from request line
1	type_code	40 (bound)
2	source_instal	from call to clock_
3	object_instal	from call to clock_
7	area_use	from request line
10	inferior_list	component names added by the action of other requests
11	path_list	
11.0	source_path	"source_path" (command) "bound_name" (request)
11.1	object_path	from command
11.2	old_dir	from command
11.3	info_dir	from command

All other items remain null.

Usage:

```
call msl_add$bound (msl_path, source_path, object_path,
                    old_dir, info_dir);
```

1) msl_path(char(*)) name of MSL to be edited

2) object_path(char(*)) MSL entry item

3) old_dir(char(*)) MSL entry item

4) info_dir(char(*)) MSL entry item