

Published: 10/30/69

Identification

msl\_transmog  
Edwin W. Meyer, Jr.

Purpose

A modification to an MSL (Multics Segment List) or LSM (List Structure Manipulator - see MSPM section BY.22) procedure can cause an MSL format change such that an existing MSL is incompatible with the new MSL procedures.

The procedure msl\_transmog is used to transmogrify (convert) an MSL from an existing format to the new format. It can also be used to coalesce two or more MSL's, to extract dead storage from an MSL, or to repair certain types of damage in an MSL.

Usage

```
call msl_transmog (msl_path, new_msl_path);
```

- 1) msl\_path(char(\*))            pathname of MSL to be transmogrified
- 2) new\_msl\_path(char(\*))        pathname of MSL to be created or added to

msl\_transmog initiates the old MSL at "msl\_path" and attempts to initiate an existing MSL at "new\_msl\_path". If it can not be found, a virgin MSL is created at "new\_msl\_path".

Each entry in the old MSL is examined and an identical entry in the new MSL is created. If the entry in the new MSL already exists it is overwritten with all non-null items of the same entry in the old MSL.

Two links must be made for msl\_transmog to work:

msl\_old to the bound msl procedures utilizing the existing format

msl\_new to the bound msl procedures utilizing the new format

Implementation

msl\_transmog passes over the entry list of the old MSL, copying all items of each entry. An entry is copied by issuing calls to entries of msl\_util and msl\_util1 in bound segment msl\_old to pick up the entry items as character strings (each defined by a pointer and length). The entry items are stored in the new MSL by calls to entries of msl\_util and msl\_util1 in bound segment msl\_new.

msl\_transmog is able to transform an MSL into any new format, assuming that the format change is transparent to the entries and permitted sequence of usage of msl\_util and msl\_util1.