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.

<u>Usage</u>

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".

Fach 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

<u>Implementation</u>

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_until1 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.