

System Library Manager User and Customization Guide

Version 1.01

Revised 12/11/2002

Help! My changes to SYS1.PARMLIB are not active.

Did you use SLM?

No. Why?

It would have told you that SYS1.PARMLIB is the 3rd library in our PARMLIB concatenation and that the member you updated is actually being used out of the 1st library.

Oh!

Author:

Lionel B. Dyck
Kaiser Permanente Information Technology
25 N. Via Monte Ave
Walnut Creek, California 94598
e-mail: lionel.b.dyck@kp.org

No warranty or guarantee is provided with this software application. This software is licensed under the GNU Public License (GPL), a copy of which can be found at <http://www.gnu.org/licenses/gpl.html>.

Table of Contents

Overview	4
Why use SLM?	4
Why multiple System PARMLIBs	4
Concepts used by SLM	5
Configuration Options	5
Member Displays	5
Member Saving	6
Save-To Dataset	6
Member Moving	6
Using SLM	7
Add Library Option	10
Insert Library Option	11
Panel Commands	13
ACTIVE	13
ALL	13
BROWSE	13
DUPLICATE	13
EDIT	13
FILTER	13
LOCATE	14
NEW	14
SORT	14
REFRESH	14
REPORT	15
Member Selection Line Commands	16
BROWSE	16
COMPARE	16
DELETE	17
EDIT	17
MOVE	17
RENAME	18
VIEW	18
Installation	19
Customization variables	20
JES2	20
permsto	20
saveopt	20
saveoptu	21
savechar1 and savechar2	21
slmpds	21
stagedsn	21
stagesuf	21
Testing	22
Service and Support	22

Table of Figures

Figure 1: Sample SYSTEM PARMLIB Libraries.....	4
Figure 2: Sample ISRDDN Display	5
Figure 3: Member Display Options	5
Figure 4: Display of Available System Library Concatenations	7
Figure 5: Display of Libraries in the System PARMLIB Concatenation	7
Figure 6: Active Display of Members in the System PARMLIB Libraries.....	8
Figure 7: All Display of Members in the System PARMLIB Libraries	8
Figure 8: Display of Duplicate Members	9
Figure 9: Active Display with Filter Enabled.....	9
Figure 10: Display of Test Library Definition.....	10
Figure 11: Display of Members in Test Library.....	10
Figure 12: Example of an Added Library.....	10
Figure 13: Sample Added Library Selected.....	11
Figure 14: Library Insert Entry.....	11
Figure 15: Local Definition	12
Figure 16: Library Definition Dataset Entry	12
Figure 17: Sample REPORT Display	15
Figure 18: Sample Compare Prompt	16
Figure 19: Sample Delete Confirmation Prompt.....	17
Figure 20: Sample Rename Prompt.....	18
Figure 21: List of Members in the Installation PDS	19
Figure 22: List of Installation Datasets.....	19
Figure 23: List of options available for the SAVEOPT value	20

Overview

The System Library Manager, or SLM, is a tool for the OS/390 and z/OS systems programmer to simplify the task of updating the correct system library concatenation. If the installation has only a single system PARMLIB (e.g. SYS1.PARMLIB) then this tool will still provide some benefits, however the major benefit comes when there are multiple libraries defined for the system PARMLIB or other defined system library concatenations such as PROC libraries or JES Parameter libraries.

If SLM is called using the NOSYS keyword then it can be used by the non Systems Programmers (as the PARMLIB and the ADD JES2 options are disabled) as a simple source member manager with all the controls available to protect inactive members and make member backups.

Why use SLM?

SLM is designed to simplify managing the members of multiple, concatenated, system libraries. With multiple system libraries in a concatenation it is possible for the same member name to reside in multiple libraries with only the member in the highest level library being used. An update to a member lower in the concatenation will be ignored, with the potential that an IPL may be required to activate those changes in that member once it is moved or copied to a higher level library in the concatenation. With SLM it is impossible to update a member that is not the active member.

The philosophy behind the System Library Manager is that all changes will be made in a staging library that is the top library in the library concatenation. Once the change has been validated the updated member will be moved to the next lower level library for normal usage. Any library below these two will consist of the unmodified vendor supplied libraries. If selected members in the vendor libraries do require changes those modified members will be located in the installation libraries.

Why multiple System PARMLIBs

For over 30 years the MFT/MVT, VS1, SVS, MVS and OS/390 systems have had only a single SYS1.PARMLIB, which has worked fine in most cases. IBM, listening to customer feedback, has with OS/390 and z/OS provided the ability to have multiple system PARMLIB datasets. This allows the installation to have up to 11 PARMLIB datasets (including the default of SYS1.PARMLIB). With this the IBM, and possibly ISV, system PARMLIB datasets can be placed into the concatenation and used as is with an installation PARMLIB dataset higher in the concatenation to support over-rides.

Some examples of how one might use multiple system PARMLIBs would be:

1. Create a local PARMLIB at the top for use with over-rides needed by your local requirements, leaving the other PARMLIBs in their original state. This top level library will then contain very few members, which will make it easier to find your local customizations.
2. Create multiple local PARMLIBs with the top level being used for members being changed, while the second level is used for updates/overrides to IBM or ISV provided members, which reside in the lower level libraries.

For example, assume the following is the normal concatenation for a shop:

SYS1.CHANGE.PARMLIB	Library where all updates are made until proven
SYS1.LOCAL.PARMLIB	Library where all local over-rides exist after being proven
SYS1.PARMLIB	IBM default
CPAC.PARMLIB	Serverpac default
SYS1.VENDOR.PARMLIB	Other vendor parmlib default

Figure 1: Sample SYSTEM PARMLIB Libraries

In this example the first library, SYS1.CHANGE.PARMLIB, would be used as the staging library where all changes take place and where changed and new members are validated. Once the changed or new members are validated they would be moved to the second library, SYS1.LOCAL.PARMLIB. During the move any pre-existing member of the same name in the second library would be backed up using a backup naming convention (of which SLM supports several). This would leave the other libraries with original vendor supplied members for reference or use, if the members can be used without change.

To determine if you have concatenated system PARMLIBs, enter the command **TSO ISRDDN** from within ISPF, then **PARMLIB**, or look at your **SYS1.IPLPARM** in the active **LOADxx** member for **PARMLIB** statements. If you don't know which **LOADxx** member is active use the **Console** command **D IPLINFO**.

Using **ISRDDN** the display will look something like this:

Current Data Set Allocations					Row 61 of 101
Command ==>					Scroll ==> CSR
Volume	Disposition	Act	DDname	Data Set Name	Actions: B E V M F C I Q
PAGE1R		>	PARMLIB	SYS1.ZOS.PARMLIB	
PAGE1R		>		SYS1.PARMLIB	
IODFTN		>		SYS1.TEST.PARMLIB	
RESR03		>		SYS1.IBM.PARMLIB	

Figure 2: Sample ISRDDN Display

In this example there are four libraries in the PARMLIB concatenation. The top library is the working library for updated members until those members are proven. The second library is the primary library for local members with the next for testing purposes (a hold over from before we used multiple libraries for PARMLIB) and the last is the library provided by IBM on the SYSRES.

Concepts used by SLM

To provide flexibility for the installation, while also providing a degree of integrity, the SLM application has the following features or concepts built into it:

Configuration Options

Configuration options are defined in the **SLMCUST** REXX Exec and documented in the last section of this document. See the [Installation](#) chapter.

On each of the table display panels the **REPORT** command can be used to display the current active configuration settings.

Member Displays

There are several member displays to assist in managing the system libraries.

ACTIVE	This is the default and displays only the active members found in the System libraries. If a member exists in multiple libraries the only member displayed will be the first or top level member.
ALL	Displays all members in all libraries.
DUPLICATE	Displays only duplicate members so that it is easy to identify members that are inactive and thus will never be used.

Figure 3: Member Display Options

Sorting is allowed in all displays with the default being a sort by member name, or member name and library concatenation number. Sorting is performed using the column names.

Member Saving

When the **EDIT** or **RENAME** member commands are used the original member is saved using an installation option. For example, the original member might have the last character changed to a Z or perhaps the last 2 characters reversed. There is also an option to retain 2 or 4 levels of the original, if desired. See the **saveopt** customization variable for the various supported options.

Save-To Dataset

This feature allows the installation to select a specific library into which all updates will be made by either selecting an option during installation or by allowing the individual systems programmers to make the selection. When this feature is enabled all New members are created in the **save-to** library. When an existing member is selected with Edit and it currently resides in a lower level library in the system library concatenation, it will be copied into the **save-to** library for editing using the **stagesuf** suffix to identify the working member.

Member Moving

This option is provided to allow a member created in the **save-to** dataset to be moved into the next lower library. This allows the **save-to** dataset to be used for newly created, *unproven* members. After some period of time the members can be moved out of the staging library.

Using SLM

Once SLM is installed and customized you can begin to use it. To begin with you need to know that you start the application by using the following command from any ISPF Command line (assuming SLM is installed in the active TSO and ISPF library concatenation):

TSO %SLM option

The only supported option at this time is **NOSYS**. When **NOSYS** is used the dynamic addition of the System **PARMLIB** and the **ADD JES2** command are disabled. This allows SLM to be used by non System Programmers as a simple version manager.

The initial entry panel will look something like the panel below:

```
----- System Library Management Library List 1.01 --- Row 1 of 2
Command ==>                                     Scroll ==> CSR

Commands: Add Insert Cancel Report
or          S Select D Delete I Insert R Repeat or overtyp to change

Sel Lib-Name      Description
_   PARMLIB        System Parmlibs - dynamically determined on RSYS_____
_   TESTPARAM      SLM Test Parmlib Libraries_____
***** Bottom of data *****
```

Figure 4: Display of Available System Library Concatenations

This panel displays the system determined active PARMLIB libraries and any other list of libraries that the user defines, in this case TESTPARAM.

Upon selecting PARMLIB, for example, the next display is:

```
----- PARMLIB Dataset Options 1.01 ----- Row 1 of 4
Command ==>                                     Scroll ==> CSR

Select the library to save into or leave default to individual library
Line commands: S Select for Save-To, B Browse, or U to UnSelect the Save-To

Sel # Library                               Vol      Status
_   1 SYS1.ZOS.PARMLIB                       Save-To  Library
_   2 SYS1.PARMLIB
_   3 SYS1.TEST.PARMLIB
_   4 SYS1.IBM.PARMLIB
***** Press PF3 (End) to Continue *****
```

Figure 5: Display of Libraries in the System PARMLIB Concatenation

In this case there are four system PARMLIB libraries. The first is defined as the Save-to Library. This means that any members selected to be edited in any of the libraries will be saved into this one library. The next library is the primary PARMLIB library with the next two libraries available for read only. This allows you to have IBM (and possibly ISV) libraries in the concatenation. When you need to make an override the updated member is placed into the Save-To library, leaving the original intact.

Options here are to Select a library as the Save-to Library or to Browse the library to view the contents. If you decide to allow the member to be edited in the original library at all times then you can decide not to select a Save-To Library and you can also Un-Select the library at some future date. The key is that this

should be consistent among all the systems programmers using this dialog. If the Save-to Library is still valid then press PF3 (End) to continue to the next display:

```

----- ACTIVE      PARMLIB  Local Datasets  1.01  --- Row 1 of 437
Command ==>                               Scroll ==> CSR
Filter:
Commands   : Active All Browse Dup Edit Filter Locate New Refresh Report Sort
Line Options: B Browse C Compare D Delete E Edit M Move R Rename V View

S Name      Lib   VV MM  Created  Changed      Size  Init   Mod  ID
_ $$$COIBM  4     14 00  02/02/06 02/02/06 15:43    0     0    0  SCLM
_ ADYSET02  2     01 00  96/12/18 96/12/18 11:05    15    15    0  SYSSJW1
_ ADYSET00  2     01 00  96/12/18 96/12/18 11:05    15    15    0  SYSSJW1
_ ADYSET01  2     01 01  99/09/02 99/09/07 12:44    11    11    0  SYSLBD
_ ADYSET02  2
_ ALLOC00   2     01 00  94/12/01 94/12/01 17:35   143   143    0  SYSSJW1
_ ANTXIN00  4
_ APPCPMA1  2     01 00  02/03/28 02/03/28 15:54     8     8     0  SYSMXB
_ APPCPMA2  2     01 01  02/03/28 02/03/28 15:54     8     8     0  SYSMXB
_ APPCPMA3  2     01 00  02/03/29 02/03/29 10:24     8     8     0  SYSMXB
_ APPCPMDL  2     01 01  98/01/17 98/01/17 13:50     2     2     0  SYSSAWA
_ APPCPMD1  2     01 00  02/03/28 02/03/28 15:55     6     6     0  SYSMXB
_ APPCPMD2  2     01 01  02/03/28 02/03/28 15:55     6     6     0  SYSMXB
_ APPCPMD3  2     01 01  02/03/29 02/03/29 10:25     6     6     0  SYSMXB
_ APPCPM00  2     01 02  98/07/21 98/07/22 11:46    14    14    0  SYSEMT

```

Figure 6: Active Display of Members in the System PARMLIB Libraries

In this display (which is using sample libraries) all the **active** members from the active system PARMLIBs are displayed. If a member also resides in a library lower in the concatenation it will not be displayed. This display can be refreshed by entering the **RE**fresh or **A**ctive command. You can limit the display by using the **F**ILTER command.

If the **ALL** command is used the following is displayed:

```

----- ALL      PARMLIB  Local Datasets  1.01  --- Row 3 of 504
Command ==>                               Scroll ==> CSR
Filter:
Commands   : Active All Browse Dup Edit Filter Locate New Refresh Report Sort
Line Options: B Browse C Compare D Delete E Edit M Move R Rename V View

S Name      Lib   VV MM  Created  Changed      Size  Init   Mod  ID
_ ADYSET00   1     01 00  96/12/18 02/11/21 09:58    15    15    0  SYSLBD
_ ADYSET00  2     01 00  96/12/18 96/12/18 11:05    15    15    0  SYSSJW1
_ ADYSET00  4
_ ADYSET01   2     01 01  99/09/02 99/09/07 12:44    11    11    0  SYSLBD
_ ADYSET01  4
_ ADYSET02   2
_ ADYSET02  4
_ ALLOC00    2     01 00  94/12/01 94/12/01 17:35   143   143    0  SYSSJW1
_ ANTXIN00   4
_ APPCPMA1   2     01 00  02/03/28 02/03/28 15:54     8     8     0  SYSMXB
_ APPCPMA2   2     01 01  02/03/28 02/03/28 15:54     8     8     0  SYSMXB

```

Figure 7: All Display of Members in the System PARMLIB Libraries

Notice the **ADYSET00** members. Only the first, in library 1, will be used by the system while the other copies, in libraries 2 and 4, are effectively hidden. Without a tool like this in place it is possible that updates could be made to the member in a library other than 1 without realizing that such an update will never be active.

The **Duplicate** command is illustrated in Figure 8 and demonstrates how easy it is to identify members of the defined library concatenation that will never be accessed using the normal system library access process.

```

----- D U P L I C A T E   P A R M L I B   L o c a l   D a t a s e t s   1.01   --- Row 1 of 132
Command ==>                                         Scroll ==> CSR
  Filter:
Commands      : Active All Browse Dup Edit Filter Locate New Refresh Report Sort
Line Options: B Browse C Compare D Delete E Edit M Move R Rename V View

S Name      Lib  VV MM  Created  Changed          Size  Init  Mod  ID
_ ADYSET00  2    01 00  96/12/18 96/12/18 11:05    15    15    0  SYSSJW1
_ ADYSET00  4
_ ADYSET01  2    01 01  99/09/02 99/09/07 12:44    11    11    0  SYSLBD
_ ADYSET01  4
_ ADYSET02  2
_ ADYSET02  4
_ ASAIPCSP  2
_ ASAIPCSP  4
_ ASBIPCSP  2    01 00  99/09/03 99/09/03 08:11    44    44    0  SYSLBD
_ ASBIPCSP  4
_ ATBIPCSP  2
_ ATBIPCSP  4
_ BLSCECT   2    01 00  96/06/18 96/06/18 10:01   486   486    0  SYSSJW
_ BLSCECT   4
_ BLSCECTX  2    01 00  93/04/02 93/04/02 14:01    84    84    0  SYSSJW
_ BLSCECTX  4

```

Figure 8: Display of Duplicate Members

The next figure illustrates the use of the **FILTER** command to limit the display to just those members that begin with the characters IEASYS. It is **STRONGLY** recommended that the filter never be more than 6 characters so that the 2 character suffix is not hidden.

```

----- A C T I V E   P A R M L I B   L o c a l   D a t a s e t s   1.01   ----- Row 1 of 5
Command ==>                                         Scroll ==> CSR
  Filter: IEASYS*
Commands      : Active All Browse Dup Edit Filter Locate New Refresh Report Sort
Line Options: B Browse C Compare D Delete E Edit M Move R Rename V View

S Name      Lib  VV MM  Created  Changed          Size  Init  Mod  ID
_ IEASYSAS  2    01 00  02/02/04 02/02/04 14:35    63    63    0  SYSAPS
_ IEASYSSB  2    01 03  01/12/31 01/12/31 19:17     3     5     0  SYSSEB
_ IEASYSZ3  2    01 06  02/11/05 02/11/13 12:57     6     6     0  SYSJWT
_ IEASYS00  2    01 11  02/01/08 02/11/13 11:27    61    63    0  SYSJWT
_ IEASYS91  2    01 00  01/11/26 01/11/26 15:25    63    63    0  SYSSEBA
***** End of Member List *****

```

Figure 9: Active Display with Filter Enabled

The next two figures demonstrate the sample TESTPARM library definition. Figure 10 displays the datasets defined to this library. Notice that three datasets are defined and the first is defined as the Save-Lib, which is the dataset in which all working members will be created and edited in.

```

----- System Library Management TESTPARM 1.01 Row 1 to 3 of 3
Command ==> Scroll ==> CSR

Enter or Verify datasets to use and then PF3 (End) to Continue

Commands: Insert or Insert dsn Report
or D Delete I Insert R Repeat or ovrtype to change

Sel Dsname Volume Save-Lib
_ 'SYSLBD.SLM.TEST.PARM0' _____ Y
_ 'SYSLBD.SLM.TEST.PARM1' _____ -
_ 'SYSLBD.SLM.TEST.PARM2' _____ -
***** Press PF3 (End) to Continue *****

```

Figure 10: Display of Test Library Definition

Figure 11 below displays the sample members of these libraries. There is one member, IEASYS0\$, that is currently being updated but has not yet been renamed to be active. This member resides in the Save-Lib dataset, which is the top level dataset in the concatenation. The active IEASYS00 is in the second dataset and it has four backup copies (as the **saveopt** value of 6 was being used).

```

----- ACTIVE TESTPARM Local Datasets 1.01 Row 1 to 6 of 6
Command ==> Scroll ==> CSR
Filter:
Commands : Active All Browse Dup Edit Filter Locate New Refresh Report Sort
Line Options: B Browse C Compare D Delete E Edit M Move R Rename V View

S Name Lib VV MM Created Changed Size Init Mod ID
_ IEASYS0$ 1 01 22 02/11/08 02/11/20 14:34 56 56 0 SYSLBD
_ IEASYS00 2 01 22 02/11/08 02/11/19 09:02 56 56 0 SYSLBD
_ IEASYS01 2 01 22 02/11/08 02/11/19 09:02 56 56 0 SYSLBD
_ IEASYS02 2 01 22 02/11/08 02/11/18 14:33 56 56 0 SYSLBD
_ IEASYS03 2 01 22 02/11/08 02/11/18 14:30 56 56 0 SYSLBD
_ IEASYS04 2 01 22 02/11/08 02/11/18 14:30 56 56 0 SYSLBD
***** End of Member List *****

```

Figure 11: Display of Members in Test Library

Add Library Option

The **Add** command can be used to dynamically add any library concatenation that is currently allocated to the active TSO/ISPF session for the life of this invocation of SLM. The syntax is **Add ddname** as demonstrated in the following example where the active SYSEXEC library is added:

```

----- System Library Management Library List 1.01 Row 1 to 3 of 3
Command ==> Scroll ==> CSR

Commands: Add Insert Cancel Report
or S Select D Delete I Insert R Repeat or ovrtype to change

Sel Lib-Name Description
_ PARMLIB System Parmlibs - dynamically determined on ASYS_____
_ SYSEXEC SYSEXEC - dynamically determined on ASYS for SYSLBD_____
_ TESTPARM SLM Test Parmlib Libraries_____
***** Bottom of data *****

```

Figure 12: Example of an Added Library

The result is that all the current datasets allocated to SYSEXEC are added as a temporary library. When selected it displays the following.

For JES2 environments where procedure libraries are not defined dynamically the use of **JES2** as the **ddname** will cause all of the procedure libraries defined to JES2 to be added as temporary libraries to SLM.

Note that no Save-Lib is defined. A Save-Lib can be defined if desired or if left blank then Edit member selections will be processed in the active library. When there is no Save-Lib defined then the Staging suffix member will be created in the same library as the selected member and the Move selection will not be allowed.

```

----- System Library Management SYSEXEC 1.01 Row 1 to 13 of 13
Command ==> Scroll ==> CSR

Enter or Verify datasets to use and then PF3 (End) to Continue

Commands: Insert or Insert dsn Report
or         D Delete I Insert R Repeat or overtype to change

Sel  Dsname                               Volume  Save-Lib
--  -
-   'SYSLBD.LIONEL.EXEC'                   _____
-   'SYSL.ISPF.TEST.EXEC'                 _____
-   'SYSL.ISPF.KFH.EXEC'                  _____
-   'ISP.SISPEXEC.VB'                    _____
-   'SYS2.REXX.EXEC'                      _____
-   'SYS2.ISPF.USERS.EXEC'                _____
-   'RDC.ISPF.REXX.EXEC'                  _____
-   'SYS1.SBPXEXEC.VB'                    _____
-   'EOY.SEOYCLIB.VB'                     _____
-   'SYS2.OPSMVS.VBREXX'                  _____
-   'CBC.SCBCUTL.VB'                      _____
-   'REXX.V1R3M0.SFANCMD.VB'              _____
-   'REXX.V1R3M0.SEAGCMD.VB'             _____
***** Press PF3 (End) to Continue *****

```

Figure 13: Sample Added Library Selected

Insert Library Option

The **Insert** command, or line selection, will bring up the following insertion prompting panel:

```

----- Library Definition Entry 1.01 -----
Command ==>

Library Name      ==> local
Library Description ==> My Local Libraries

```

Figure 14: Library Insert Entry

You will notice in the following example that the library was added in sorted sequence and the Library Name was converted to all caps.

```

----- System Library Management Library List 1.01 Row 1 to 4 of 4
Command ==> Scroll ==> CSR

Commands: Add Insert Cancel Report
or S Select D Delete I Insert R Repeat or overtype to change

Sel Lib-Name Description
- LOCAL My Local Libraries
- PARMLIB System Parmlibs - dynamically determined on ASYS
- SYSEXEC SYSEXEC - dynamically determined on ASYS for SYSLBD
- TESTPARM SLM Test Parmlib Libraries
***** Bottom of data *****

```

Figure 15: Local Definition

Then Select the new library to display the next panel where you can enter the individual library names:

```

----- System Library Management LOCAL 1.01 Row 1 to 1 of 1
Command ==> Scroll ==> CSR

Enter or Verify datasets to use and then PF3 (End) to Continue

Commands: Insert or Insert dsn
or D Delete I Insert R Repeat or overtype to change

Sel Dsname Volume Save-Lib
-----
***** Press PF3 (End) to Continue *****

```

Figure 16: Library Definition Dataset Entry

On this panel just enter the dataset name and optional volume in the empty row. If you want to define the Save-Lib then enter a Y on that row. To add more rows use the Insert Command or Insert Selection command.

The Insert command syntax is:

```

Insert data-set-name
or Insert number-of-rows (default is 1 row)

```

If the dataset name is entered without quotes, the quotes will be added if required (e.g. SYS1.PARMLIB should be entered as 'SYS1.PARMLIB'). This is a key stroke saving option for entering dataset names.

Panel Commands

ACTIVE

Returns the member display to the normal display after using the **ALL** command.

ALL

Displays **all** members in all libraries sorted by name and library order.

Note: When using this option the ability to create new members using the **NEW** command is restricted, as is the use of the Edit line command.

BROWSE

Browse the specified member.

e.g. **B IEASYS00** or **S IEASYS00**

DUPLICATE

Displays all duplicate members in all the concatenated System PARMLIB libraries.

Note: Use of **ACTIVE**, **ALL**, **DUPLICATE**, **FILTER**, or **REFresh** will also refresh the current display.

EDIT

Edit the specified member.

e.g. **E IEASYS00**

FILTER

Filter will limit the display to members that match the filter specification. The filter may be a specific member name or a member mask (a member naming ending with an *). If the member name specified is less than 8 characters then an * is automatically appended for member name filtering.

For example, to display only the IEASYSxx members enter:

FILTER IEASYS*

It is **STRONGLY** recommended that the filter not be more than 6 characters to prevent backup members from being hidden.

LOCATE

To locate a member use the **L xxx** command where xxx is the member name or a member name prefix.

Examples:

To find member IEASYS00 enter:

```
L IEASYS00
```

To find the first IEAIPS member enter:

```
L IEAIPS
```

NEW

To create a new member use the NEW command followed by the new member name:

```
NEW IEASYS01
```

The member will be created in either the Save-To-Library, or if there is no Save-To-Library defined, it will be created in the top library in the concatenation list. If the requested member name already exists an error message will be displayed.

SORT

Allows the table to be sorted using any of the column header names. The syntax is:

```
SORT Field-name,Order
```

Where order is A for ascending (the default) or D for descending.

Examples:

```
SORT CREATED
```

```
SORT CREATED LIB
```

```
SORT CREATED ,D  
(sort by creation date in descending order)
```

REFRESH

Refreshes the current member display.

REPORT

Displays a report of the currently active customization settings as demonstrated in the example below.

```
System Library Manager Current Settings 12/11/02 12:52:15  
Table Library in use : 'SYSLBD.slmpps'  
Staging Suffix      : $  
Save Character 1    : Z  
Save Character 2    : Y  
Staging PARMLIB     :  
PARMLIB Save Library : 1  
JES2 PROC Name      : JES2  
Save Option In Use  : Save backup changing last character to Z
```

Figure 17: Sample REPORT Display

Member Selection Line Commands

BROWSE

To browse a member enter a **B** in the S (for selection) column.

COMPARE

To compare a member enter a **C** in the S (for selection) column. A prompting panel will be displayed to request the specification of a dataset name, member, and optional volser. The ISPF Compare utility will be invoked and the results displayed in Browse. If necessary you can specify whether to compare with or without sequence number checking.

Below is an example of the Compare prompt. Notice that you can select any one dataset from the list of datasets for this set of libraries, or you can enter any fully qualified dataset of your choice.

```
----- PARMLIB Compare Dataset 1.01 ----- Row 1 of 4
Command ==>

Compare member: IEASYS00   in dataset: SYS1.PARMLIB
to Dataset      : _____
to Member       : IEASYS00   if blank will use Compare Member name
to Volume       : _____ optional
Sequence Numbers: No       Yes or No

Enter the dataset name and optional volume serial, or select the dataset
from the list below. Then press enter to Compare or PF3 to cancel.

Sel #Dsname                                     #Volume
-----
- 'SYS1.ZOS.PARMLIB'
- 'SYS1.PARMLIB'
- 'SYS1.TEST.PARMLIB'
- 'SYS1.IBM.PARMLIB'
***** End of Member List *****
```

Figure 18: Sample Compare Prompt

DELETE

- To Delete a member enter a **D** in the S (for selection) column.
- If the member is **NOT** the **ACTIVE** member then Delete will not be allowed.

Below is an example of the prompting that occurs when a Delete is requested.

```
----- Delete Verification 1.01 -----  
Command ==>  
  
Confirm delete of member $$$COIBM  
from dataset: SYS1.IBM.PARMLIB  
volume:  
  
Press Enter to Confirm or PF3 to Cancel
```

Figure 19: Sample Delete Confirmation Prompt

EDIT

To edit a member enter an **E** in the S (for selection) column.

What will occur at this point depends on the Save-To option:

- If the Save-To Library is different from the library in which the selected member resides then the selected member will be copied into the Save-To Library and then placed into ISPF Edit based on the staging suffix that was defined for the installation. If the member is not updated, that is a SAVE is not performed, the copied member will be deleted (un-copied).
- If the Save-To Library is in the same as the library in which the selected member resides, the original member will be saved in that library under a different name, based on the staging suffix that was defined at the time of the SLM installation.
- If the member is **NOT** the **ACTIVE** member then Edit will not be allowed.

MOVE

To move a member enter an **M** in the S (for selection) column. This option is used to move a member from the staging library into the next library below in the concatenation. If the member already exists in the target library then it is backed up using the backup options defined by **saveopt**. Note that a staging member that has not been renamed cannot be moved.

RENAME

- To rename a member enter an **R** in the S (for selection) column. A rename prompting panel will be displayed to request the new name for the member.
- If the member is **NOT** the **ACTIVE** member then Rename will not be allowed.

```
----- PARMLIB  Rename Dataset  1.01  -----  
Command ==>  
  
Rename member:  IEASYS0$  in dataset: SYS1.PARMLIB  
                IEASYS00
```

Figure 20: Sample Rename Prompt

VIEW

Invokes the ISPF View function on the requested member.

Installation

To install SLM follow these simple steps:

1. Unzip the **SLM.ZIP** file into a working directory
2. Do a binary file transfer of the **SLM.XMIT** file to your OS/390 system into a sequential dataset with **RECFM=FB** and **LRECL=80** (the BLKSIZE can be any valid multiple of 80 or use system determined blocksize). This dataset will require about two 3390 tracks.
3. Assuming you uploaded the file into a dataset of the name **hlq.SLM.XMIT** (where hlq is your TSO high level qualifier – probably your userid) then issue the following command from either **TSO READY** or ISPF option 6:

RECEIVE INDS(SLM.XMIT)

When prompted press the Enter key to take the default dataset name or enter **DS(dsname)** to define a different dataset name for the partitioned dataset to be created.

4. You now have a dataset of the name **hlq.SLM.PDS** unless you changed the dataset name. This dataset contains the following members:

Member	Description
\$\$DOC	Brief documentation on the members and how to use them
\$RECEIVE	Sample exec to expand the members into usable datasets
CHANGES	Change history
EXEC	REXX library in TSO Transmit format
LICENSE	The GPL License under which this application is distributed
LOAD	LOAD library in TSO Transmit format
MSGS	ISPF MSGS library in TSO Transmit format
PANELS	ISPF Panels library in TSO Transmit format

Figure 21: List of Members in the Installation PDS

5. To get the TSO Transmit members into usable datasets issue the following command from **TSO READY** or ISPF Option 6:

EX 'hlq.SLM.PDS(\$RECEIVE)' EX

(the final EX is to tell the Execute command that you are executing a REXX EXEC)

6. You will now have the following datasets available:

Dataset	Description
hlq.SLM.NEW.EXEC	REXX library with the SLM execs and a test exec
hlq.SLM.NEW.LOAD	LOAD library with the SLMF REXX Function load module
hlq.SLM.NEW.MSGS	ISPF Messages for the dialog
hlq.SLM.NEW.PANELS	ISPF Panels for the dialog

Figure 22: List of Installation Datasets

Next customize the installation by editing the **hlq.SLM.NEW.EXEC** member **SLMCUST** to review and/or update the variables in the code for your installation.

Customization variables

Customization occurs in the **SLMCUST** REXX Exec that will be found in the same library as the **SLM** REXX Exec. Making the customizations requires only a minimal level of knowledge of REXX coding syntax.

JES2

JES2 is used to define the name of the JES2 PROC which will be processed if the ADD JES2 command is used.

permsto

permsto is used to define a specific library in the system library concatenation for the system PARMLIB that will always be used to save the edited members. This allows an installation to **prevent** other libraries from being updated and places all updates in a specific library. If this value is null then the user is allowed to select a library for this purpose from the list of available libraries when the application starts.

For example, in the list of libraries that is displayed in **Figure 5**, the first library is where all changes are placed until they are proven and the other libraries are not updated directly. To enforce this you would set **permsto** to 1. Be aware that if the **stagedsn** value is defined then **permsto** is automatically set to that library regardless of what is coded and the user cannot change it.

saveopt

saveopt is used to define how SLM will save the original member when EDIT is selected:

Value	Definition
1	Reverse the last 2 characters of the member name. e.g. IEASYS0K is backed up as IEASYSK0
2	Change the 2 nd to last character of the member name to the character defined in variable savechar1 e.g. IEASYS0K is backed up as IEASYSZK (assuming savechar1 is set to Z)
3	Change the last 2 characters to 99 in the backup member name. e.g. IEASYS0K is backed up as IEASYS99
4	Save 2 copies of the original. The oldest member has the last character changed to the character defined in savechar2 and the most recent saved with the last character changed to the character defined in savechar1 . e.g. when IEASYS0K is edited member IEASYS0Z would be renamed to IEASYS0Y and IEASYS0K would be saved as IEASYS0Z. This assumes savechar1 is Z and savechar2 is Y .
5	Same as 4 except the next to last character is changed. e.g. when IEASYS0K is edited member IEASYSYK would be renamed to IEASYSZY and IEASYS0K would be saved as IEASYSYK. This assumes savechar1 is Z and savechar2 is Y .
6	Save 4 levels of the member by renaming the member from xxx0 to xxx1 to xxx2 to xxx3 to xxx4.
anything else	Creates a backup member with the last character changed to the character defined in savechar1 . e.g. IEASYS0K would be backed up as IEASYS0Z if savechar1 were defined as Z .

Figure 23: List of options available for the SAVEOPT value

saveoptu

saveoptu is defined using the same options available to **saveopt** and is used when SLM is started with the **NOSYS** option. This allows an installation to have a different save methodology for the systems programmers and another for the non systems programmers.

savechar1 and savechar2

savechar1 and **savechar2** are used to define the characters that will be used in creating the backup member names when a member of the system library is edited. The defaults are:

- **savechar1** = Z
- **savechar2** = Y

slmpds

slmpds is used to define the library where the System Library Manager will save any custom library lists (ISPF Tables) that are defined. If the user defined library concatenations are not going to be shared then set this to:

```
'''sysvar('syspref')'.slmpds''
```

Note that this library is used exclusively by each user, so while you may define this library with a common high level qualifier, the library will be enqueued such that only a single user can use it at a time.

stagedsn

stagedsn is used to define a library that is **not** in the active system library concatenation into which all updates will be saved. The updated members can then be copied into production at a later date, after they have been verified in this staging library using any locally developed process. This library will appear as the top/first library in the library display and members within will appear as the active member in the member selection table display. By default this value is null.

stagesuf

stagesuf is used to indicate that you wish to only create (New) or Edit members with this suffix. This is a single character (e.g. \$) which when not null causes the following actions:

- On New member creation the member name is created with the last character of the **stagesuf** (e.g. the command NEW IEASYSAA will actually result in member IEASYSAA\$ being created)
- On Edit the member is copied into a member name with the last character changed to the **stagesuf** (e.g. editing member IEASYS00 will result in member IEASYS00\$ being created and opened in edit).

This feature is provided to allow an installation to create staging members within the existing system libraries, either in the defined save-to (see **permsto**) or in the library where the member resides. The **rename** line command can then be used to rename the newly created member to the desired member name with the original member being saved based upon the **saveopt** specification.

Testing

To test this application copy member **TESTSLM** for into a library in your **SYSPROC** or **SYSEXEC** concatenation or issue the following command from ISPF Option 6 to test it:

```
EX 'hlq.SLM.NEW.EXEC(TESTSLM)'
```

Once you are happy that the application works as you want then you can continue to use it via the TESTSLM exec or you can copy the individual distributed libraries into appropriate production libraries in your environment (the recommended method) and invoke it by the **TSO %SLM** command.

Service and Support

The SLM package is maintained and supported on a time available basis. For mutual support there is a web based group and listserv available.

Information on it can be found at:

Home Page	http://groups.yahoo.com/group/system_library_manager/
Post message:	system_library_manager@yahoogroups.com
Subscribe:	system_library_manager-subscribe@yahoogroups.com
Unsubscribe:	system_library_manager-unsubscribe@yahoogroups.com