

SLFLIB
TP8 Librarian
USER MANUAL
Version 2.3A

Copyright 1990-1999 by
B & C Solution - Denmark
Revision date: September 1999

PREFACE

This document introduces and explains the operational and installation procedures of SLFLIB - the efficient tool to manage System-Loadable File and TP8 TPR libraries within the GCOS-8 environment.

The manual has been organized in 6 chapters.

Chapter 1 gives a general introduction.

Chapter 2 describes the SLFLIB directives.

Chapter 3 describes the SLFLIB operation procedures.

Chapter 4 describes the SLFLIB output reports and messages.

Chapter 5 describes the ARUED directives.

Chapter 6 describes the ARUED operation procedures.

Chapter 7 describes the ARUED output reports and messages.

Chapter 8 describes the installation/update procedure.

Chapter 9 describes the software problem reporting procedure.

B & C Solution makes no representations or warranties with respect to the contents hereof and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Further, B & C Solution reserves the right to make changes from time to time without obligation of B & C Solution to notify any person of such revision or changes.

Suggestions and criticisms concerning the form, content and purpose of this manual are invited and should be forwarded to:

B & C Solution
Ole Piisvej 4, Box 54
DK-3100 Hornbaek
DENMARK

Telephone: +45 49.70.47.90
Facsimile: +45 49.70.47.80
E-mail: support@bc-solution.dk

TABLE OF CONTENTS

1. General Introduction

- 1.1 LODLIB Compatibility
- 1.2 TP-LIB Compatibility

2. SLFLIB Directives

- 2.1 General Description
 - 2.1.1 Standard Library Directives
 - 2.1.2 TP8 Library Directives
- 2.2 CLEAR Directive
- 2.3 APPEND Directive
- 2.4 DELETE Directive
- 2.5 REPLACE Directive
- 2.6 GET Directive
- 2.7 GETFS Directive
- 2.8 SEARCH Directive
- 2.9 UPDATE Directive
- 2.10 TABLE Directive
- 2.11 GATE Directive
- 2.12 GROUP Directive
- 2.13 PRIMARY Directive
- 2.14 SECONDARY Directive
- 2.15 STAMP Directive

3. SLFLIB Operation Procedures

- 3.1 General Description
 - 3.1.1 Example - Batch Mode JCL
 - 3.1.2 Example - TP8 Mode JCL
 - 3.1.3 Example - GROUP Mode JCL
- 3.2 Batch Mode Operation
 - 3.2.1 Example - Creation of Library
 - 3.2.2 Example - Maintenance of Library
- 3.3 TP8 Mode Operation
 - 3.3.1 Example - TP8 TPR Library Maintenance
 - 3.3.2 Example - TP8 Workstation and Commands
 - 3.3.3 Example - TP8 TPR Library Synchronization
 - 3.3.4 Use of BIBO
- 3.4 GROUP Mode Operation
 - 3.4.1 Example - TP8 GROUP TPR Library Maintenance
 - 3.4.2 Another example - TP8 GROUP TPR Library Maintenance
- 3.5 SLFLIB Update Technique
- 3.6 SLFLIB P* Handling

4. SLFLIB Output Reports and Messages

- 4.1 Directive Listing Report
 - 4.1.1 Batch Mode
 - 4.1.2 TP8 Mode
 - 4.1.3 GROUP Mode
- 4.2 Directory Listing Report
 - 4.2.1 LOCATION Sequence
- 4.3 Error Messages
 - 4.3.1 Information Messages
 - 4.3.2 Warning Messages
 - 4.3.3 Fatal Error Messages

5. ARUED Directives

- 5.1 General Description
- 5.2 GATE Directive
- 5.3 PRIMARY Directive
- 5.4 SECONDARY Directive
- 5.5 TABLE Directive

6. ARUED Operation Procedures

- 6.1 General Description
- 6.2 TP8 Mode Operation
 - 6.2.1 Example - TP8 Virtual TPR Library Maintenance
 - 6.2.2 Example - TP8 Virtual TPR Library Synchronization
 - 6.2.3 Use of BIBO
- 6.3 ARUED Update Technique
- 6.4 ARUED P* Handling

7. ARUED Output Reports and Messages

- 7.1 Directive Listing Report
- 7.2 Directory Listing Report
- 7.3 Error Messages
 - 7.3.1 Warning Messages
 - 7.3.2 Fatal Error Messages

8. Installation/Update Procedures

- 8.1 General description
- 8.2 Software Diskette Procedure
- 8.3 Software Tape Procedure
- 8.4 Software Customization & Generation
 - 8.4.1 Software Customization
 - 8.4.2 Software Generation
- 8.5 Software Integration
 - 8.5.1 PRMFL Reference
 - 8.1.2 INSERT-EDIT
- 8.6 Software Release Bulletin

- 8.6.1 Warnings
- 8.6.2 Problems fixed
- 8.6.3 Changes
- 8.6.4 Enhancements
- 8.6.5 Migration

9. Software Problem Reporting

- 9.1 General Description
- 9.2 Software Problem Report
- 9.3 Software Problem Distribution
- 9.4 Software Release Documentation

Chapter 1

General Introduction

The System-Loadable Library utility program - SLFLIB, is designed to create and maintain system-loadable program libraries. The program library may be used to execute program elements through JCL or as a TPR library for TP8.

SLFLIB provides various library maintenance functions:

- initialization of library
- random insertion, deletion and update of program(s)
- program directory listing (sorted)
- TP8 library group synchronization
- TP8 library group selection

SLFLIB supports both System-Loadable (LODLIB) format and DMIV TP-LIB format libraries.

The Advanced Run-unit Library utility program - ARUED, is designed to manage virtual TPR libraries, and provide synchronization functionality similar to SLFLIB. ARUED does not update the Run-unit Library, instead it controls the execution of one or more RUED activities. The RUED utility is part of UW-Tools provided by BULL.

1.1 LODLIB Compatibility

As SLFLIB is intended as a replacement for the standard LODLIB utility, the library formats are completely compatible as is most of the functionality. The few non-supported features of LODLIB has been considered of limited value and/or very seldom used features, where as the added features of SLFLIB address the requirements of a System-Loadable File Librarian in the current GCOS-8/TP8 environment.

Following features of LODLIB are not supported:

- The directives PATCH, PEND, FILNAM and HEAD are not supported.
- Output to tape (Total System) is not supported.
- The .GPAED member, related to PATCH directive, is treated as a normal member and no patch maintenance is performed should such member exist.

Following information is relevant when both SLFLIB and LODLIB are used to update library files:

- SLFLIB maintains a date and time stamp for each program, whenever a program is inserted or updated on the library file. The stamp is used both for documentation and synchronization control in the TP8 librarian mode. Although the library file may be updated both from LODLIB and SLFLIB, care must be taken when using LODLIB, since it erases all stamps from the library file. Stamps may be re-inserted with the STAMP directive to ensure proper synchronization in the TP8 librarian mode.
- When using LODLIB on a library file last updated with SLFLIB, the length specification in the directory (TABLE) is incorrect, due to SLFLIB's use of date and time stamp. This error is automatically corrected by LODLIB for subsequent runs and has no practical impact.
- SLFLIB creates a dummy member (S.L.F.) on the library file for stamp maintenance. A LODLIB directory will include this dummy member otherwise it has no practical impact.

1.2 TP-LIB Compatibility

With the re-introduction of DMIV TP-LIB for the TP8 environment (IT3.0), SLFLIB has been extended to include both LODLIB and TP-LIB format support. SLFLIB naturally maintains compatibility with TP-LIB library format, however the functional operation of a TP-LIB library, is equivalent for both TP-LIB and LODLIB format. Thus SLFLIB is not compatible with any form of TP-LIB directive syntax or JCL procedures.

Following features of SLFLIB are NOT available with TP-LIB Librarian:

- Automatic library group synchronization.
- Use of TP-LIB libraries as input for library update.
- Library re-organization (re-build).
- Automatic library group selection.

Chapter 2

SLFLIB Directives

This chapter describes the syntax, argument options and the operation of the SLFLIB directives.

2.1 General Description

Directives may begin in any column and are terminated by a blank. The operand separators are blanks or commas depending on the directive. At least one blank must be used to separate the directive from the operand. Optional operands are in brackets [...]. Only columns 1 through 72 are processed.

2.1.1 Standard Library Directives

CLEAR [TPLIB]	Initialize library.
APPEND member-name[,...]	Insert new member(s) in library.
DELETE member-name[,...]	Delete member(s) from library.
REPLACE member-name[,...]	Replace existing member(s) in library.
GET fc[,fc]	Declare additional input files for library update.
GETFS userid/ [catalog[\$password]/].. filename[\$password] [fc]	Declare additional input files for library update by using permfile description.
SEARCH userid/ [catalog[\$password]/].. filename[\$password] [fc]	Declare additional input files for library update by specifying a list of permfile descriptions.
UPDATE	Force an automatic update of the library.
TABLE [member-name[,...]] [-sequence]	List sorted directory of member(s) in library.

2.1.2 TP8 Library Directives

GATE userid/ [catalog[\$password]/].. filename[\$password]	Declare gate for concurrent TP8 library update by using permfile description.
GROUP group-id	Define a TP8 library group.
PRIMARY userid/ [catalog[\$password]/].. filename[\$password]	Declare primary TP8 library by using permfile description.
SECONDARY userid/ [catalog[\$password]/].. filename[\$password]	Declare secondary TP8 library by using permfile description.
STAMP [date [time]]	Insert date and time stamp on all members in library.

2.2 CLEAR Directive

The CLEAR directive initializes the System-Loadable File library. SLFLIB supports two different library formats - LODLIB and TP-LIB. The library file format is defined at the time of initialize and cannot change afterwards (except during new initialize). Default is LODLIB format, which is standard for both TP8 and batch libraries, whereas TP-LIB only pertain to TP8.

Note: Maximum library file size is 52.428 llinks for LODLIB format and 108.000 llinks for TP-LIB format.

2.2.1 Syntax

```
CLEAR [TPLIB or TP8LIB]
```

2.2.2 Rules

- The CLEAR directive must precede any other directives when a new library is used.
- The CLEAR directive may not be used in TP8 mode. Instead each library file may be initialized separately in normal batch mode.
- TPLIB argument may be used to request TP-LIB format (default is LODLIB format). For TP8 IT4.1, use TP8LIB argument to ensure compatibility.
- The CLEAR directive is not immediately processed by LODLIB. An error message is issued when such deferred CLEAR from LODLIB is detected by SLFLIB. The CLEAR directive must be processed through SLFLIB, before any attempt to update the library file through SLFLIB.

Note: Any subsequent CLEAR directive will initialize the entire library and all members will be lost.

2.3 APPEND Directive

The APPEND directive causes a new member or members to be appended to the library from the update file.

2.3.1 Syntax

```
APPEND member-name[,...]
```

2.3.2 Rules

- Member-name identifies the name of the member on the update file (input).
- If the member-name already exists in the library, the directive is ignored and an error message is issued.
- If the member-name contains the mask character (*), the remaining part of the member-name will be ignored, thus all members starting with the identification up to the * are appended, ex. ABC*.
- Each member-name specified will be processed separately.

2.4 DELETE Directive

The DELETE directive causes an existing member or members to be deleted from the library file.

2.4.1 Syntax

```
DELETE member-name[,...]
```

2.4.2 Rules

- The member-name identifies the member on the library file to be deleted.
- If the member-name does not exist in the library file, the delete request is ignored and an error message is issued.
- If the member-name contains the mask character (*), the remaining part of the member-name will be ignored, thus all members starting with the identification up to the * are deleted, ex. ABC*.

- Each member-name specified will be processed separately.

2.5 REPLACE Directive

The REPLACE directive causes an existing member or members to be deleted from the library file and new member or members to be appended to the library from the update file.

2.5.1 Syntax

```
REPLACE member-name[,...]
```

2.5.2 Rules

- Member-name identifies both the member on the library file and the member on the update file.
- If the member-name does not exist in the library file, the replace request is ignored and an error message is issued.
- If the member-name contains the mask character (*), the remaining part of the member-name will be ignored, thus all members starting with the identification up to the * are replaced, ex. ABC*.
- Each member-name specified will be processed separately.

2.6 GET Directive

The GET directive declares one or more update files (input) for subsequent handling of APPEND, REPLACE and UPDATE directives.

2.6.1 Syntax

```
GET fc[,...]
```

2.6.2 Rules

- The GET directive must precede directives referencing program elements from the update file(s) (input) allocated to the specified file code(s).
- Following file-codes are reserved by SLFLIB and may not be used in the GET directive: I*, P*, PP, SC, OT, QX, QT, QG and #A through #Z.

- SLFLIB reads its input from the file designated by the file code and places its content on the library update file, for subsequent process of APPEND, REPLACE and UPDATE directives.
- A GET directive without any preceding APPEND, REPLACE or UPDATE directive between this and the previous GET, GETFS or SEARCH directive, will cause an automatic update from the previous content of the library update file.
- When multiple input files are specified in one directive, whether GET or SEARCH, these are concatenated and treated as one input file read into the library update file.

Note (LODLIB format only): Update files must be LODLIB format, if the library file is in LODLIB format.

2.7 GETFS Directive

The GETFS directive declares update file (input) for subsequent handling of APPEND, REPLACE and UPDATE directives. It has the same functionality as the GET directive, however input file may be specified through a permfile description, allowing dynamic allocation.

2.7.1 Syntax

```
GETFS userid/[catalog$password]/filename$password] [fc]
```

2.7.2 Rules

- The GETFS directive must precede directives referencing program elements from the update file (input) declared by the specified permfile description.
- A file code may optionally be specified, otherwise the file will be allocated by SLFLIB using an arbitrary file code in the range #A through #Z. Following file-codes are reserved by SLFLIB and may not be used in the GETFS directive: I*, P*, PP, SC, OT, QX, QT, QG and #A through #Z.
- SLFLIB reads its input from the file designated by the permfile description and places its content on the library update file, for subsequent process of APPEND, REPLACE and UPDATE directives.
- A GETFS directive without any preceding APPEND, REPLACE or UPDATE directive between this and the previous GET, GETFS or SEARCH directive, will cause an automatic update from the previous content of the library update file.

2.8 SEARCH Directive

The SEARCH directive declares one or more update files (input) for subsequent handling of APPEND, REPLACE and UPDATE directives. It has the same functionality as a series of GETFS directives.

2.8.1 Syntax

```
SEARCH userid/[catalog[$password]/]filename[$password] [fc]
```

2.8.2 Rules

- The SEARCH directive must precede directives referencing program elements from the update file(s) (input) declared in the permfile description(s) in the search rules file.
- The search rules file is a system standard format (SSF) file in BCD. Each record on the file is treated as a permfile description with an optional file code, as follows:

```
permfile description [fc]
```

A maximum of 24 permfile descriptions may be specified without file code or as many as required, provided the file codes are unique within the search rules file.

- A file code may optionally be specified, otherwise the file will be allocated by SLFLIB using an arbitrary file code in the range #A through #Z. Following file-codes are reserved by SLFLIB and may not be used in the SEARCH rules file: I*, P*, PP, SC, OT, QX, QT, QG and #A through #Z.
- SLFLIB reads its input from the file(s) designated by the permfile description of each record in the search rules file and places its content on the library update file, for subsequent process of APPEND, REPLACE and UPDATE directives.
- A SEARCH directive without any preceding APPEND, REPLACE or UPDATE directive between this and the previous GET, GETFS or SEARCH directive, will cause an automatic update from the previous content of the library update file.
- When multiple input files are specified in one directive, whether GET or SEARCH, these are concatenated and treated as one input file read into the library update file.

2.9 UPDATE Directive

The UPDATE directive forces an automatic update of all members on the library update file to the library file.

2.9.1 Syntax

UPDATE

2.9.2 Rules

- An automatic update is equivalent to a series of APPEND or REPLACE directives, referencing the entire content of the library update file.
- Under certain conditions an automatic update is forced. (See rules under GET, GETFS and SEARCH.) Automatic update is also forced when the last directive is a GET, GETFS or SEARCH directive.
- If arguments are specified, the UPDATE request is ignored and an error message is issued.

2.10 TABLE Directive

The TABLE directive produces a sorted directory listing of the library file. The sorting sequence may be controlled through the sequence argument. And implicit library verification process is invoked when selecting the -Location sorting sequence.

2.10.1 Syntax

```
TABLE [member-name[,...]] [-N[ame]] [-D[ate]] [-L[ocation]]
```

2.10.2 Rules

- If member-name(s) is not specified the directory for the entire library is listed.
- If the member-name contains the mask character (*), the remaining part of the identification is ignored, thus a directory of all members starting with the identification up to the * are listed, ex. ABC*.
- Sequence argument -NAME (default) produces directory listing sorted on ascending member name.
- Sequence argument -DATE produces directory listing sorted on descending date and time for last update of member.
- Sequence argument -LOCATION produces directory listing sorted on ascending library location. In addition to member information, the directory listing contain information on following library elements:

- Dirblk - Directory block
- Spablk - Space-Directory block
- Free - Free block

An implicit library verification process is performed to check for overlapping blocks or non-reflected holes in the library. Any discrepancies are reported and an error message is issued.

- Each directory entry contains the member-name, entry point, size in words, offset, relocation, master, location (block number), size in blocks, date and time of the last update. All numeric fields except date and time are presented in octal form.

2.11 GATE Directive

The GATE directive declares a common dummy resource file for protection against concurrent update of TP8 libraries.

2.11.1 Syntax

```
GATE userid/[catalog[$password]/]filename[$password]
```

2.11.2 Rules

- The designated permfile description is exclusively allocated to SLFLIB during library update, to prevent concurrent update of TP8 libraries.
- A default gate permfile description (CMDLIB/SLFLIB/GATE) is used when no GATE directive is specified.
- When present the GATE directive must be the first directive specified in the input file.
- A GATE directive will set SLFLIB into TP8 mode.

2.12 GROUP Directive

The GROUP directive declare a library group for TP8.

2.12.1 Syntax

```
GROUP group-id
```

2.12.2 Rules

- The group-id (1-8 characters) provides a logical identification of a TP8 library group. This identification is used by the Group Selection Interface module (SLFGSI), which process SITE specific rules based on the group-id, ref. Chapter 5 for details.
- The GROUP directive must be the first or second directive specified in the input file, depending on the presence of a GATE directive.
- A GROUP directive will set SLFLIB into GROUP mode, which also imply TP8 mode.
- A maximum of 10 group's may be declared.

2.13 PRIMARY Directive

The PRIMARY directive declares the primary TPR library file for TP8 or the primary TPR library file within a TP8 library group.

2.13.1 Syntax

```
PRIMARY userid/[catalog[$password]/]filename[$password]
```

2.13.2 Rules

- The designated permfile description is used to allocate the primary TPR library file for TP8 mode or GROUP mode operation.
- The PRIMARY directive must be the first or second directive specified in the input file, depending on the presence of a GATE directive, or immediate follow any GROUP directive.
- A PRIMARY directive will set SLFLIB into TP8 mode, unless SLFLIB is already set to GROUP mode.

Note: Primary and Secondary TPR library files must have the same format, either LODLIB or TP-LIB.

2.14 SECONDARY Directive

The SECONDARY directive declare the secondary TPR library file for TP8 or the secondary TPR library file within a TP8 library group.

2.14.1 Syntax

```
SECONDARY userid/[catalog[$password]/]filename[$password]
```

2.14.2 Rules

- The designated permfile description is used to allocate the secondary TPR library file for TP8 mode operation.
- The SECONDARY directive must immediately follow the PRIMARY directive.
- The PRIMARY and SECONDARY declarations have no special meaning others than identification of the two library files.

2.15 STAMP Directive

The STAMP directive forces a date and time stamp to be applied to all members in the library file.

2.15.1 Syntax

```
[STAMP [date [time]]]
```

2.15.2 Rules

- Format of date is YYMMDD.
- Format of time is HHMMSS.
- When time or date and time is not specified, current date and/or time will be used as default.
- The STAMP directive may be used (through batch mode operation) to re-establish stamp information on the library file. History information however is lost since the same (latest) stamp will be applied to all members. TP8 mode operation may be resumed when both PRIMARY and SECONDARY library files have been properly stamped.

Note (LODLIB format only): The date and time stamp is critical for TP8 mode operation. If library file(s) are updated through LODLIB, these stamps are lost and synchronization becomes impossible. An error message is issued, when such condition is detected in TP8 mode and the execution is aborted. Since stamps are maintained both in a dummy member and in each member entry, the error message includes information on the latest stamp applied to the library file.

Chapter 3

SLFLIB Operation Procedures

This chapter describes how JCL procedures are set-up or amended for utilization of SLFLIB to maintain System-Loadable File Libraries, and how the unique TP8 Librarian features of SLFLIB are utilized.

3.1 General Description

A System-Loadable File Library may be used for storage of multiple programs in System-Loadable format - also referred to as H* or Q* format. Programs may be executed from the library through standard JCL:

```
$ IDENT      .....
$ PROGRAM MYPROG
$ PRMFL     **,R,R,MYUMC/MYLIB
```

The System-Loadable format is created by the General Loader, which provide an option for save of the program (LOAD-only) as opposed to direct execution (LOAD-and-Go).

```
$ IDENT      .....
$ OPTION   CBL74, ...,SAVE/MYPRG
$ CBL74
... source deck of program ...
$ EXECUTE
$ FILE     H*,X1S,10L (System-Loadable File)
```

The H*-file is a System-Loadable File Library containing only the one member MYPRG. Multiple members in the same H*-file may be maintained, either implicit if program contains overlays or segmentation or explicit through the use of SAVOLD, which may be used to maintain multiple programs one the same H*-file.

The use of SLFLIB or standard LODLIB provides a more flexible way of System-Loadable File Library maintenance. The maintenance is performed through an additional activity, which use the H*-file created by the General Loader.

```
...
$ FILE     H*,X1S,10L
$ LODLIB
$ PRMFL    OT,W,R,MYUMC/MYLIB
Diirectives .....
$ FILE     fc,X1R
```

SLFLIB is an enhanced version of LODLIB, which may be installed (ref. chapter 5) to replace LODLIB or as an optional product. When installed as optional, which is assumed in the following examples, the LODLIB activity must include a reference to the SLFLIB program file CMDLIB/SLFLIB/HSTAR/LODLIB.

SLFLIB provides three operation modes:

Batch Mode - which is very similar to standard LODLIB.

TP8 Mode - which provides dynamic handling of TP8 TPR libraries.

GROUP Mode - which provides dynamic selection of TP8 TPR libraries.

3.1.1 Example - Batch Mode JCL

```

$      IDENT    .... as required ...
$      LODLIB
$      PRMFL   **,R,R,CMDLIB/SLFLIB/HSTAR/LODLIB
$      PRMFL   OT,W,R, ... library ...
Directives .....
Optional files for input:
$      PRMFL   fc,R,R, ... system-loadable file ...

```

3.1.2 Example - TP8 Mode JCL

```

$      IDENT    .... as required ...
$      LODLIB
$      PRMFL   **,R,R,CMDLIB/SLFLIB/HSTAR/LODLIB
[GATE ...]
PRIMARY ...
SECONDARY ...
Directives .....
Optional files for input:
$      PRMFL   fc,R,R, ... system-loadable file ...

```

3.1.3 Example - GROUP Mode JCL

```

$      IDENT    .... as required ...
$      LODLIB
$      PRMFL    **,R,R,CMDLIB/SLFLIB/HSTAR/LODLIB
[GATE ...]
GROUP A
PRIMARY ...
SECONDARY ...
GROUP B
PRIMARY ...
SECONDARY ...
Directives .....

Optional files for input:

$      PRMFL    fc,R,R, ... system-loadable file ...

```

3.2 Batch Mode Operation

The following examples illustrate the use of SLFLIB in batch mode operation.

3.2.1 Example - Creation of library

```

$      IDENT    ... as required
$      FILSYS
CF USER/MY-LIB,BLOCKS/100,1000/,MODE/RAND/
$      LODLIB
$      PRMFL    **,R,R,CMDLIB/SLFLIB/HSTAR/LODLIB
$      PRMFL    OT,W,R,USER/MY-LIB
CLEAR
GET    F1
APPEND MYPRG1,MYPRG2
TABLE MYPRG*
$      PRMFL    F1,R,R,USER/F1-FILE
$      ENDJOB

```

The library (MY-LIB) is created and initialized, the program elements MYPRG1 and MYPRG2 is inserted from the input file F1 and a directory of all library members starting with MYPRG is listed.

3.2.2 Example - Maintenance of Library

```

$      IDENT    ... as required
$      LODLIB
$      PRMFL    **,R,R,CMDLIB/SLFLIB/HSTAR/LODLIB
$      PRMFL    OT,W,R,USER/MY-LIB
DELETE MYPRG1
GET    F1
UPDATE
GET    F2
APPEND MYPRG9
GETFS  USER/HSTAR1
SEARCH USER/RULES
TABLE
$      PRMFL    F1,R,R,USER/F1-FILE
$      PRMFL    F2,R,R,USER/F2-FILE
$      ENDJOB

```

Content of USER/RULES file:

```

USER/HSTAR2
USER/HSTAR3

```

Delete MYPRG1 from library, update library with all program elements on F1-file, append new MYPRG9 program element from F2-FILE, update (automatic) library with all program elements on USER/HSTAR1 file, update (automatic) library with all program elements on USER/HSTAR2 and USER/HSTAR3 files and produce directory listing for entire library.

3.3 TP8 Mode Operation

Dynamic online maintenance of TPR libraries may be achieved by defining a TPR library GROUP in the TP8 workstation. When enabling the TP8 workstation only the first library file is allocated, leaving the other available for batch update. Subsequently new TPR versions may be made available to TP8 through the library group ROTATE command.

The TP8 mode of SLFLIB specifically addresses the problem of determination of available TPR library file and automatic synchronization of the two TPR library files.

TPR libraries are identified via the PRIMARY and SECONDARY directives. SLFLIB will automatically determine which library file is available for update and perform update operation according to given directives (APPEND, DELETE, REPLACE and UPDATE).

Before the update SLFLIB will also determine whether the most recently update TPR library is currently in use by TP8 and perform a synchronization of the two library files if required. Synchronization merely updates the older library with changes

since its last update. The automatic synchronize operation is documented on the directive listing report from SLFLIB. A deferred synchronize technique is used when both library files are available for update. Synchronization is based on date and time stamp inserted by SLFLIB during each update.

A gate (dummy file) is used to control concurrent update from batch. A separate gate file should be used for each set of library files.

The following examples illustrate the use of SLFLIB in TP8 mode operation.

3.3.1 Example - TP8 TPR Library Maintenance

```
$      IDENT    ... as required
..
..  Compile and load TPR
..
$      FILE     H*,H01S,5R
$      LODLIB
$      PRMFL    **,R,R,CMDLIB/SLFLIB/HSTAR/LODLIB
$      FILE     10,H01R
GATE      USER/MY-GATE
PRIMARY   USER/MY-LIB1
SECONDARY USER/MY-LIB2
GET 10
UPDATE
$      ENDJOB
```

Update the TPR library group with a new TPR version.

3.3.2 Example - TP8 Workstation and Commands

Library Group definition in TP8 workstation:

```

CREATE_GLOBAL_FILE      10          &
  -WS_NAME              w-tp8      &
  -PATH_NAME            user/my-lib1 &
  -VERSION              0000       &
  -PERMISSION           R          &
  -ALLOCATION            REQUIRED     &
  -MODE                RANDOM      &
  -TYPE                LIBRARY     &
  -LIBRARY_TYPE        PUBLIC or DMIV ;

CREATE_GLOBAL_FILE      11          &
  -WS_NAME              w-tp8      &
  -PATH_NAME            user/my-lib2 &
  -VERSION              0000       &
  -PERMISSION           R          &
  -ALLOCATION            REQUIRED     &
  -MODE                RANDOM      &
  -TYPE                LIBRARY     &
  -LIBRARY_TYPE        PUBLIC or DMIV ;

CREATE_GROUP_CODE       t1          &
  -WS_NAME              w-tp8      &
  -VERSION              0000       &
  -PROGRAM_LIBRARY     YES        ;

CREATE_GROUP_LOCAL_FILE 10          &
  -WS_NAME              w-tp8      &
  -VERSION              0000       &
  -GLOBAL_FILE          10         &
  -PERMISSION           R          &
  -GROUP_CODE           t1         ;

CREATE_GROUP_LOCAL_FILE 11          &
  -WS_NAME              w-tp8      &
  -VERSION              0000       &
  -GLOBAL_FILE          11         &
  -PERMISSION           R          &
  -GROUP_CODE           t1         ;

```

Note: The LIBRARY-TYPE must match the library file format - PUBLIC for LODLIB format or DMIV for TP-LIB format.

Master Command to switch library:

```
$FILE -MODIFY ^TL
or
$FILE -MODIFY _LG (switch all library groups)
```

Note: Inclusion of new TPR's and/or TPR's changed to another library are subject to various TP8 rules and may require special action. Details on such rules are described in TP8 documentation (DH36).

3.3.3 Example - TP8 TPR Library Synchronize

```
$ IDENT ... as required
$ LODLIB
$ PRMFL **,R,R,CMDLIB/SLFLIB/HSTAR/LODLIB
GATE USER/MY-GATE
PRIMARY USER/MY-LIB1
SECONDARY USER/MY-LIB2
$ ENDJOB
```

Above example will force TPR library synchronization. This activity is recommended as part of the TP8 workstation ENABLE JCL, in order to assure complete synchronization of libraries, before startup of TP8.

3.3.4 Use of BIBO

The library group rotate (\$FILE -M) may be performed through a BIBO job. This approach is very practical for test environments, where library update activities are frequent. When using BIBO following guidelines should be followed:

- Allocate the GATE file(s) to the BIBO activity (via PRMFL) to avoid any conflict with updates during library rotate.
- Include an SLFLIB synchronize activity as second activity, to avoid confusion if a new rotate is requested, before any library updates takes place.

3.4 GROUP Mode Operation

Dynamic library group selection may be used to perform update of multiple library group's using the same JCL. Each library group is treated like described under TP8 Mode.

Group selection rules are SITE specific and SLFLIB provides a Group Selection Interface module (SLFGSI), which must be customized with these rules. Rules may

be based on TPR name, parts of TPR name and/or the size of the TPR. Rules will thus test various conditions and match this with the current group identification to determine a SELECT/NON-SELECT status. Rules may later be revised and SLFLIB will automatically adjust library content according to the new rules, i.e. DELETE a TPR from one library group and APPEND it to another.

Following example of SITE specific rules are used to illustrate how GROUP mode may be used:

SITE have three library group's:

Group A - contains all TPR's named A* or B* (* is a mask)

Group B - contains all TPR's > 100KW

Group C - contains all other TPR's

3.4.1 Example - TP8 GROUP TPR Library Maintenance

```

$      IDENT    ... as required
..
..  Compile and load TPR
..
$      FILE     H*,H01S,5R
$      LODLIB
$      PRMFL    **,R,R,CMDLIB/SLFLIB/HSTAR/LODLIB
$      FILE     10,H01R
GATE      USER/MY-GATE
GROUP A
PRIMARY   USER/MY-LIB1
SECONDARY USER/MY-LIB2
GROUP B
PRIMARY   USER/MY-LIB3
SECONDARY USER/MY-LIB4
GROUP C
PRIMARY   USER/MY-LIB5
SECONDARY USER/MY-LIB6
GET 10
UPDATE
$      ENDJOB

```

SLFLIB will use the SITE specific rules and perform maintenance of any or all of library group's involved, and synchronize the library files within each group accordingly.

3.4.2 Another example - TP8 GROUP TPR Library Maintenance

SITE have four library group's:

Group TEST - contains TPR's which are currently under development or maintenance

Group A - contains all TPR's named A* or B* (* is a mask)

Group B - contains all TPR's > 100KW

Group C - contains all other TPR's

The JCL procedure to update within the TEST group will use TP8 Mode, since this library group will be used to hold ANY TPR which is currently under development or maintenance.

```
$      IDENT    ... as required
..
..  Compile and load TPR
..
$      FILE     H*,H01S,5R
$      LODLIB
$      PRMFL    **,R,R,CMDLIB/SLFLIB/HSTAR/LODLIB
$      FILE     10,H01R
GATE      USER/MY-GATE
PRIMARY   USER/MY-TLIB1
SECONDARY USER/MY-TLIB2
GET 10
UPDATE
$      ENDJOB
```

As development or maintenance activities are completed, the TPR(s) are moved from the TEST group into the other library group. The JCL procedure to accomplish this would be as follows:

```

$      IDENT    ... as required
$ C. Move TPR(s) from TEST
$      LODLIB
$      PRMFL    **,R,R,CMDLIB/SLFLIB/HSTAR/LODLIB
$      PRMFL    10,R,R,USER/MY-TLIB1
GATE      USER/MY-GATE
GROUP A
PRIMARY   USER/MY-LIB1
SECONDARY USER/MY-LIB2
GROUP B
PRIMARY   USER/MY-LIB3
SECONDARY USER/MY-LIB4
GROUP C
PRIMARY   USER/MY-LIB5
SECONDARY USER/MY-LIB6
GET 10
REPLACE TPR1
REPLACE TPR2
$ C. Remove TPR(s) from TEST
$      LODLIB
$      PRMFL    **,R,R,CMDLIB/SLFLIB/HSTAR/LODLIB
GATE      USER/MY-GATE
PRIMARY   USER/MY-TLIB1
SECONDARY USER/MY-TLIB2
GET 10
DELETE TPR1
DELETE TPR2
$      ENDJOB

```

3.5 SLFLIB Update Technique

SLFLIB was designed to provide secure, flexible and fast System-Loadable File Library maintenance procedures.

When creating a random library file the CLEAR directive is used to format the library according to the initial size. The statistics from each SLFLIB run will indicate how much space (links) are available for insertion of new program members. As new members are inserted SLFLIB will automatically grow the library file up to the maximum file size specified. Should this be inadequate, simply adjust the maximum file size and re-run the SLFLIB job.

When inserting new or replacing existing program members in the library, SLFLIB will place the member content most efficiently. Typically even a very active library will contain only few free space slots which are unusable. The statistics will indicate the both the actual number of free space slots as well as the number of free space slots, which are below the average member size. This does not mean the space is

unusable, but less likely usable than other free space slots. The library size will not decrease automatically, however excess space may be regained by creating a new smaller library file and re-inserting the program members again from the old library file (GET + UPDATE).

SLFLIB always use a "Safely-Update" technique, which will secure the integrity of the library file should any failure occur (abort, crash etc.). Interrupted SLFLIB activities may simply be re-run after any failure.

3.6 SLFLIB P* Handling

The P* may, as with standard GCOS-8, be assigned to a PRMFL for further process or deferred printing. SLFLIB will issue an immediate FA-abort or a deferred WA-abort if errors are detected during directive processing, however technique conflicts with the most recent GCOS-8 releases, since the P* PRMFL is overwritten with GCOS-8 history dump information.

To avoid such problem, SLFLIB will instead write reports to file-code PP, when a PRMFL allocation of PP is recognized. This technique only affects JCL when a PRMFL is assigned - no SYSOUT PP is required.

```
$ PRMFL PP,W,S,... P* Output file ...
```


Chapter 4

SLFLIB Output Reports and Messages

4.1 Directive Listing Report

SLFLIB always produce a directive listing and process report on report code 74. Any errors detected are flagged with error messages together with the directive causing the error. Each message is prefixed with a class and reference number. Three classes of messages may occur:

I-nnn - Information only message
W-nnn - Warning message
F-nnn - Fatal error message

The end of the directive listing report contains basic statistical information on library utilization.

PSW Switch-35 is set ON whenever a warning or error message is given during execution. Either a warning or error message indicates that the library(s) may not have been successfully update, and switch-35 may be used to alter the behavior of your JCL.

4.1.1 Batch Mode

The UPDATE directive activates an implicit number of APPEND and/or REPLACE directives. The directive listing report will reflect this as special heading and footing lines are inserted with an information line for each implicit directive.

Following illustrate the content of the directive listing report:

```
CLEAR
XXXXX
W-017 Unrecognizable directive
GET 10
UPDATE
***** AUTOMATIC UPDATE DOES THE FOLLOWING; *****
- APPEND TPR1
- APPEND TPR2
***** END OF AUTOMATIC UPDATE *****

Library type           H*           Library group           N/A
Library size in llinks 120       Number of members      2
Available space in llinks 10       Number of catalogs     1
Available space blocks 1         Blocks below avg. size 0
```

Note (LODLIB format only): "Available space blocks" indicate how many holes the "Available space in llinks" is spread across. The "Blocks below avg. size" indicate how many of the "Available space blocks" are of a size below the average member size in the library. This indicates that such space is less likely to be re-used in subsequent update.

4.1.2 TP8 Mode

In TP8 mode the library updated is dynamically selected at execution time. The Directive listing report contains information on the library selected.

The automatic library synchronize feature may activate an implicit number of APPEND, REPLACE and/or DELETE directives. The directive listing report will reflect this as special heading and footing lines are inserted with an information line for each implicit directive.

Following illustrate the content of the directive listing report:

```

PRIMARY MYTP8/LIB1
SECONDARY MYTP8/LIB2
GET 10
UPDATE

Library used for update MYTP8/LIB1 - Last updated 91-04-01 08:49:00

***** AUTOMATIC UPDATE DOES THE FOLLOWING; *****
- APPEND TPR3
***** END OF AUTOMATIC UPDATE *****

Library type           H*      Library group          N/A
Library size in llinks 180    Number of members     3
Available space in llinks 10    Number of catalogs    1
Available space blocks  1     Blocks below avg. size 0

Library used for input MYTP8/LIB1 - Last updated 91-04-01 09:01:02
Library used for update MYTP8/LIB2 - Last updated 91-04-01 08:49:15

***** AUTOMATIC SYNCHRONIZE DOES THE FOLLOWING; *****
- APPEND TPR3 91-04-01 09:01:02

Library type           H*      Library group          N/A
Library size in llinks 180    Number of members     3
Available space in llinks 10    Number of catalogs    1
Available space blocks  1     Blocks below avg. size 0
***** END OF AUTOMATIC SYNCHRONIZE *****
    
```

4.1.3 GROUP Mode

In GROUP mode, which imply TP8 mode also, both the library group and library file updated are dynamically selected at execution time. The Directive listing report contains information on the library group and library file selected.

The automatic library synchronize feature may activate an implicit number of APPEND, REPLACE and/or DELETE directives. The directive listing report will reflect this as special heading and footing lines are inserted with an information line for each implicit directive.

Following illustrate the content of the directive listing report:

```
GROUP A
PRIMARY MYTP8/LIB1
SECONDARY MYTP8/LIB2
GROUP C
PRIMARY MYTP8/LIB3
SECONDARY MYTP8/LIB4
GET 10
UPDATE

Library used for update MYTP8/LIB1 - Last updated 94-04-01 08:49:00
***** AUTOMATIC UPDATE DOES THE FOLLOWING; *****
- APPEND ABC1
***** END OF AUTOMATIC UPDATE *****

Library used for update MYTP8/LIB3 - Last updated 94-04-01 08:49:00
***** AUTOMATIC UPDATE DOES THE FOLLOWING; *****
- APPEND TPR3
***** END OF AUTOMATIC UPDATE *****
```

```

Library used for input MYTP8/LIB1 - Last updated 94-04-01 09:01:02
Library used for update MYTP8/LIB2 - Last updated 94-04-01 08:49:15

***** AUTOMATIC SYNCHRONIZE DOES THE FOLLOWING; *****
- APPEND ABC1 94-04-01 09:01:02

Library type           H*           Library group           A
Library size in llinks 180 Number of members      3
Available space in llinks 10 Number of catalogs      1
Available space blocks  1  Blocks below avg. size 0
***** END OF AUTOMATIC SYNCHRONIZE *****

Library used for input MYTP8/LIB4 - Last updated 94-04-01 09:01:02
Library used for update MYTP8/LIB3 - Last updated 94-04-01 08:49:15

***** AUTOMATIC SYNCHRONIZE DOES THE FOLLOWING; *****
- APPEND TPR3 94-04-01 09:01:02

Library type           DMIV          Library group           C
Library size in llinks 180 Number of members      3
Available space in llinks 10 Number of catalogs      1
Available space blocks  1  Blocks below avg. size 0
***** END OF AUTOMATIC SYNCHRONIZE *****

Library type           H*           Library group           A
Library size in llinks 180 Number of members      3
Available space in llinks 10 Number of catalogs      1
Available space blocks  1  Blocks below avg. size 0

Library type           DMIV          Library group           C
Library size in llinks 180 Number of members      3
Available space in llinks 10 Number of catalogs      1
Available space blocks  1  Blocks below avg. size 0
    
```

4.2 Directory Listing Report

The TABLE directive produces a member directory listing report on report code 01. For GROUP mode operation, a directory listing is produced for each group and the group identification is inserted in the heading line. The sorting sequence is controlled through the sequence argument, which provide following three options:

- NAME - Sorted ascending on member-name
- DATE - Sorted descending on date and time for last update
- LOCATION - Sorted ascending on library location

Each directory entry contains following information:

- Name - Name of member
- Entry - Program entry point (octal)

Size	- Size of program in words (octal)
Offset	- Offset to beginning of program (octal)
Reloc	- Relocatable object indication
Master	- Master indication
#Blk	- Start library block for program (octal)
#Blks	- Number of library blocks for program (octal)
Date/Time	- Date and time for last update (YY-MM-DD HH.MM.SS)

Note: A library block is one 64-word sector = 1/5 llink.

4.2.1 LOCATION Sequence

The directory listing report for LOCATION sequence will contain further directory entries, which document the actual utilization of the complete library file. Such directory entries are:

Dirblk	- Directory block (15 members per block)
Spablk	- Space-Directory block (62 free entries per block)
Free	- Free block(s)

An implicit library verification process is automatically performed to check for overlapping or non-reflected holes in the library file. Any discrepancies are reported directly in the directory listing report and an error message is issued on the directive listing report also.

4.3 Error Messages

When SLFLIB detect any error conditions during operation, either an immediate (FATAL) or deferred (WARNING) abort occur, thus the status of an SLFLIB activity may be checked as follows:

NORMAL TERMINATION - All directives processed.

ABORT-CODE "WA" - One or more directives not processed.

ABORT-CODE "FA" - Fatal error detected during process.

Directives are processed one by one and any update of the library file is permanent after a directive has been processed error free. A fatal error abort will only impact the current and following directives, which are not processed.

4.3.1 Information Messages

I-001 The library is empty

```
I-002 Last stamp used by SLFLIB YY-MM-DD HH:MM:SS
```

This message is issued after the F-004 message and may be used to construct a proper STAMP directive.

```
I-003 The library 'XX' is empty
```

```
I-004 The library is NOT a multiple of 12 llinks
```

Your current TP-LIB format library is not compatible with standard TP-LIB.

4.3.2 Warning Messages

```
W-001 Syntax error during directive process - ignored
```

```
W-002 UPDATE directive - arguments not allowed
```

```
W-003 Cannot find member 'XXXXXX' in library
```

```
W-004 Member 'XXXXXX' is not defined in above GET/GETFS/SEARCH
```

An APPEND or REPLACE directive reference a member-name which was not located on the UPDATE file produced by preceding GET/GETFS/SEARCH directives.

```
W-005 No members fits this name 'XXXXXX'
```

A group of members were referenced using the * mask character, however no members were found within the masked group.

```
W-006 The library 'XX' is empty
```

```
W-007 CLEAR directive must precede all editing directives
```

```
W-008 CLEAR needed for 'XXXXXX' file - LODLIB has been used
```

A CLEAR directive must be processed through SLFLIB, since LODLIB CLEAR handling is deferred and not reflected on the library file. Re-run SLFLIB with a CLEAR directive and re-run the SLFLIB activity again.

```
W-009 Input filecode 'XX', attempted access beyond file - ignored
```

Check for input file 'XX' for proper format, ex. failed GELOAD activity.

```
W-010 Editing directives must be preceded by GET, GETFS, SEARCH
```

```
W-011 Input file 'XX' not present
```

A referenced filecode 'XX' was not allocated.

```
W-012 Maximum of 24 catalog/file strings exceeded
```

```
W-013 Member 'XXXXXXX' already exists in library
```

```
W-014 Member 'XXXXXXX' Filecode 'XX' is null - ignored
```

```
W-015 Library too fragmented - should re-build
```

SLFLIB will maintain a maximum of 1000 "Available space blocks". Further fragmentation will cause available space to be lost, however this does not impact the integrity of the library, unless a TABLE -LOCATION directive is performed. The library file should be re-build to reorganize the available space, since excessive fragmentation degrades performance.

```
W-016 Error in SEARCH - file has bad media code
```

SEARCH rule files must be System-Standard-Format with BCD code-set.

```
W-017 Unrecognizable directive
```

```
W-018 Waiting - Both PRIMARY and SECONDARY libraries are busy
```

```
W-019 Unknown or misplaced directive
```

GATE, GROUP, PRIMARY and SECONDARY directives must be given in a predefined sequence, ref. chapter 2.

4.3.3 Fatal Error Messages

```
F-001 Library format bad (Space Directory)
```

The library file has incorrect format. Check the library allocation for errors. Report problem and try to restore and re-run SLFLIB activities again or re-build a new library (using corrupted library as input) as a temporary solution.

```
F-002 Library format bad (Directory)
```

The library file has incorrect format. Check the library allocation for errors. Report problem and try to restore and re-run SLFLIB activities again as a temporary solution.

```
F-003 Error occured during write to UPDATE file 'QX'
```

Permanent I/O error on temporary file 'QX'. Check for disk problems on device.

```
F-004 LODLIB has been used on 'XXXXXX' file - STAMP directive required
```

An update with LODLIB has erased time-stamp information for all members. Run SLFLIB with the STAMP directive one file at a time to re-establish time-stamps and re-run SLFLIB activity again.

```
F-005 Error in date/time format: STAMP YYMMDD HHMMSS
```

```
F-006 Library file 'OT' not allocated
```

The library file 'OT' not allocated and/or TP8 mode not detected.

```
F-007 Unable to grow library file to NNNNN llinks
```

SLFLIB failed to grow the library file. Increase maximum size to required or larger size. The maximum library size is 52.428 (LODLIB format) or 108.000 (TP-LIB format) llinks and SLFLIB cannot grow above this limit. Split library file into two separate libraries if the maximum is reached, or change to TP-LIB format.

```
F-008 Maximum library size 52.428/108.000 llinks
```

SLFLIB will not utilize a library file created bigger than 52.428 (LODLIB format) or 108.000 (TP-LIB format) llinks.

```
F-009 Both PRIMARY and SECONDARY libraries are busy
```

SLFLIB cannot detect which library to update, since neither library is available for "write" allocation. Check the library files and TP8 operation.

Note: Could be caused by incorrect use of GATE resource file also.

```
F-010 Wrong file-code 'XX'
```

Internal error - report the problem.

F-011 I/O error occurred on filecode 'XX'

Permanent I/O error on filecode 'XX'. Check for disk problems on device.

F-012 Checksum error in directory on filecode 'XX'

Check for input file 'XX' for proper format, ex. a failed GELOAD activity.

F-013 Attempted access to sector zero on filecode 'XX'

Check for input file 'XX' for proper format, ex. a failed GELOAD activity.

F-014 Library has no space catalog

The library file has incorrect format. Check the library allocation for errors. Report problem and try to restore and re-run SLFLIB activities again as a temporary solution.

F-015 Directive SECONDARY expected

SECONDARY directive must immediately follow a PRIMARY directive.

F-016 Directive PRIMARY expected

PRIMARY directive must immediately follow a GATE directive.

F-017 Cannot allocate filecode 'XX'

SLFLIB cannot dynamically allocate a file to filecode 'XX'. Check directive for catalog/file string and check the file for problems.

F-018 Preceding and following entry overlap

F-019 Space between preceding and following entry

F-020 Error detected during TABLE -LOCATION

Error(s), ref. F-018 and F-019 above, reported from the automatic LOCATION verification process. Report problem and try to re-build library as a temporary solution.

F-021 Library format bad (Access beyond)

The library file has incorrect format. Check the library allocation for errors. Report problem and try to restore and re-run SLFLIB activities again as a temporary solution.

```
F-022 Allocation of XXXXXXXX failed due to YYYYYY
```

Dynamic allocation of XXXXXXX file failed. Cause indicated in YYYYY. Check file for problems.

```
F-023 Library format bad (Cross check)
```

The library file has incorrect format. Check the library allocation for errors. Report problem and try to restore and re-run SLFLIB activities again as a temporary solution.

```
F-024 Library format bad (Checksum directory)
```

The library file has incorrect format. Check the library allocation for errors. Report problem and try to restore and re-run SLFLIB activities again as a temporary solution.

```
F-025 Library format bad (Checksum space-dir)
```

The library file has incorrect format. Check the library allocation for errors. Report problem and try to restore and re-run SLFLIB activities again as a temporary solution.

```
F-026 Library size too small (truncated)
```

The library file may have been only partially copied from another library file, since directory and/or space directory reference areas beyond the file extent.

Note: SLFLIB will recover, if the missing part of the library file only pertain to free-space, otherwise results may be unpredictable. A new library may be build using the truncated library as input. Any missing TPR's will be indicated through error messages.

F-027 Can't update LODLIB format from TP-LIB format

SLFLIB will accept both LODLIB and TP-LIB input files, however TP-LIB input files may only be used to update a TP-LIB format library.

F-028 PRIMARY/SECONDARY library must have same format

TP8-Mode library files must have the same format.

F-029 Max. GROUP's (10) exceeded

A maximum of 10 groups may be defined.

F-030 Initial library size must be multiple of 12 llinks

A TP-LIB format library must be created with a size in LINKS (multiple of 12 llinks).

Chapter 5

ARUED Directives

This chapter describes the syntax, argument options and the operation of the ARUED directives.

5.1 General Description

Directives may begin in any column and are terminated by a blank. The operand separators are blanks or commas depending on the directive. At least one blank must be used to separate the directive from the operand. Optional operands are in brackets [...]. Only columns 1 through 72 are processed.

GATE userid/ [catalog[\$password]/].. filename[\$password]	Declare gate for concurrent TP8 library update by using permfile description.
PRIMARY userid/ [catalog[\$password]/].. filename[\$password][NEWEST]	Declare primary TP8 library by using permfile description.
SECONDARY userid/ [catalog[\$password]/].. filename[\$password][NEWEST]	Declare secondary TP8 library by using permfile description.
TABLE [-DATE]	List sorted directory of members in the library.

5.2 GATE Directive

The GATE directive declares a common dummy resource file for protection against concurrent update of TP8 libraries.

5.2.1 Syntax

```
GATE userid/[catalog[$password]/]filename[$password]
```

5.2.2 Rules

- The designated permfile description is exclusively allocated to ARUED during library update, to prevent concurrent update of TP8 libraries.
- A default gate permfile description (CMDLIB/SLFLIB/GATE) is used when no GATE directive is specified.

- When present the GATE directive must be the first directive specified in the input file.

5.3 PRIMARY Directive

The PRIMARY directive declares the primary Virtual TPR library file within a TP8 library group.

5.3.1 Syntax

```
PRIMARY userid/[catalog[$password]/]filename[$password] [NEWEST]
```

5.3.2 Rules

- The designated permfile description is used to allocate the primary Virtual TPR library file.
- The PRIMARY directive must be the first or second directive specified in the input file, depending on the presence of a GATE directive.
- NEWEST argument may be used to override automatic detection of newest library.

5.4 SECONDARY Directive

The SECONDARY directive declare the secondary Virtual TPR library file within a TP8 library group.

5.4.1 Syntax

```
SECONDARY userid/[catalog[$password]/]filename[$password] [NEWEST]
```

5.4.2 Rules

- The designated permfile description is used to allocate the secondary Virtual TPR library file.
- The SECONDARY directive must immediately follow the PRIMARY directive.
- NEWEST argument may be used to override automatic detection of newest library.

5.5 TABLE Directive

The TABLE directive will produce a sorted directory of members in the Virtual TPR library.

5.5.1 Syntax

```
TABLE [-DATE]
```

5.5.2 Rules

- Directory listing sorted ascending on member name (default) or descending on time of last update, when -DATE argument is given.

Chapter 6

ARUED Operation Procedures

This chapter describes how JCL procedures are set-up for utilization of ARUED, and how the unique TP8 Librarian features of ARUED are utilized.

6.1 General Description

A Virtual TPR library (Run-unit) may be maintained either with LKED, RUED or other standard utilities. Libraries must be properly initialized before they can be used with ARUED.

ARUED operation is similar to SLFLIB TP8 Mode operation, where two libraries are updated in synchronization. ARUED will automatically determine which library is available for update, and provide automatic synchronization of the two libraries.

Note: ARUED use the date and time stamp of the actual library members to determine which library is 'newest'. When deleting library members, care must be taken not to trigger a situation where remaining members leads ARUED to believe the other library is 'newest'. Make sure library deletes are done simultaneously on both libraries, or include an append/replace with 'newer' member date and time stamp.

6.2 TP8 Mode Operation

Dynamic online maintenance of Virtual TPR libraries may be achieved by defining a TPR library GROUP in the TP8 workstation. When enabling the TP8 workstation only the first library file is allocated, leaving the other available for batch update. Subsequently new TPR versions may be made available to TP8 through the library group ROTATE command.

ARUED specifically addresses the problem of determination of available Virtual TPR library file and automatic synchronization of the two Virtual TPR library files.

Virtual TPR libraries are identified via the PRIMARY and SECONDARY directives. ARUED will automatically determine which library file is available for update and perform the update operation. If both libraries are available for update, ARUED will update both libraries (instant synchronization).

Before the update ARUED will also determine whether the most recently updated Virtual TPR library is currently in use by TP8 and perform a synchronization of the

two library files if required. Synchronization merely updates the older library with changes since it's last update. The automatic synchronize operation is documented on the directive listing report from ARUED. Synchronization is based on date and time stamp inserted by LKED when the Virtual TPR was linked.

A gate (dummy file) is used to control concurrent update from batch. A separate gate file should be used for each set of library files.

The following examples illustrate the use of ARUED.

6.2.1 Example - Virtual TPR Library Maintenance

```

$      IDENT    ... as required
..
..  Compile and link TPR
..
$      FILE     V*,V1S,50R
$      PROGRAM  ARUED
$      PRMFL    **,R,R,CMDLIB/SLFLIB/HSTAR/LODLIB
GATE    USER/MY-GATE
PRIMARY USER/MY-LIB1
SECONDARY USER/MY-LIB2
$      FILE     V1,V1S
$      SELECT   CMDLIB/SLFLIB/JCL/RUED.JCL
$      ENDJOB

```

Update the Virtual TPR library group with a new TPR version or new TPR.

Note: The allocation of an Update Library with filecode V1 (must use V1S as LUD), will trigger an update of the Virtual TPR Libraries.

Use following master Command to switch (rotate) library:

```

$FILE -MODIFY ^TL

or

$FILE -MODIFY _LG (switch all library groups)

```

Note: Inclusion of new TPR's and/or TPR's changed to another library are subject to various TP8 rules and may require special action. Details on such rules are described in TP8 documentation (DH36).

6.2.2 Example - TP8 Virtual TPR Library Synchronize

```
$      IDENT    ... as required
$      PROGRAM  ARUED
$      PRMFL    **,R,R,CMDLIB/SLFLIB/HSTAR/LODLIB
GATE      USER/MY-GATE
PRIMARY   USER/MY-LIB1
SECONDARY USER/MY-LIB2
$      SELECT   CMDLIB/SLFLIB/JCL/RUED.JCL
$      ENDJOB
```

Above example will force Virtual TPR library synchronization. This activity is recommended as part of the TP8 workstation ENABLE JCL, in order to assure complete synchronization of libraries, before startup of TP8.

6.2.3 Use of BIBO

The library group rotate (\$FILE -M) may be performed through a BIBO job. This approach is very practical for test environments, where library update activities are frequent. When using BIBO following guidelines should be followed:

- Allocate the GATE file(s) to the BIBO activity (via PRMFL) to avoid any conflict with updates during library rotate.
- Include an ARUED synchronize activity as second activity, to avoid confusion if a new rotate is requested, before any library updates takes place.

6.3 ARUED Update Technique

ARUED use the RUED utility (provided with UW-Tools) to perform the actual library update activities. The ARUED JCL set-up includes 10 RUED activities, which are optionally executed after ARUED. The first 8 RUED activities are used to perform library synchronization (Append, Replace and Delete), and the last 2 RUED activities are used to update the primary and secondary Virtual TPR libraries.

Since each RUED activity can only perform a limited number of operations, and the number of RUED activities used for library synchronization is limited to eight, ARUED can be forced to perform a 'complete' synchronization. The situation will only occur when there is a substantial number of synchronization changes (Append, Replace, Delete), and merely affect the performance of the RUED operation.

6.4 ARUED P* Handling

The P* may, as with standard GCOS-8, be assigned to a PRMFL for further process or deferred printing. ARUED will issue an immediate FA-abort if errors are detected during directive processing, however technique conflicts with the most recent GCOS-8 releases, since the P* PRMFL is overwritten with GCOS-8 history dump information.

To avoid such problem, ARUED will instead write reports to file-code PP, when a PRMFL allocation of PP is recognized. This technique only affects JCL when a PRMFL is assigned - no SYSOUT PP is required.

```
$ PRMFL PP,W,S,... P* Output file ...
```

Chapter 7

ARUED Output Reports and Messages

7.1 Directive Listing Report

ARUED always produce a directive listing and process report on report code 74. Any errors detected are flagged with error messages together with the directive causing the error. Each message is prefixed with a class and reference number. Two classes of messages may occur:

W-nnn - Warning message
F-nnn - Fatal error message

The directive listing report will document the library changes - synchronization and/or update, as well as the RUED directives generated and subsequently executed.

PSW Switch-35 is set ON whenever an error message is given during execution, and all RUED activities are automatically skipped (no update will take place).

Following illustrate the content of the directive listing report, where only the secondary library is available for update and no synchronization is required:

```
GATE UMC/GATE
PRIMARY UMC/LIB1
SECONDARY UMC/LIB2

PRIMARY   library (L1) last updated 1999-08-01 12:01:00 (Read)
SECONDARY library (L2) last updated 1999-08-01 12:02:00 (Write)
UPDATE    library (V1) last updated 1999-08-15 08:15:00

***** AUTOMATIC UPDATE DOES THE FOLLOWING: *****
Update TPR1 1999-08-15 08:15:00

RUED Update=fc*L2 Include=fc*V1
***** END OF AUTOMATIC UPDATE *****
```

Following illustrate the content of the directive listing report, where the two libraries are synchronized and both libraries are updated:

```
GATE UMC/GATE
PRIMARY UMC/LIB1
SECONDARY UMC/LIB2

PRIMARY library (L1) last updated 1999-08-01 12:01:00 (Write)
SECONDARY library (L2) last updated 1999-08-01 12:02:00 (Newest,Write)
UPDATE library (V1) last updated 1999-08-15 08:15:00

***** AUTOMATIC SYNCHRONIZE DOES THE FOLLOWING: *****
Append MEMBER2 1999-08-01 12:02:00
Delete MEMBER3 1999-08-01 12:01:00
Replace MEMBER4 1999-08-01 12:02:00
Append MEMBER5 1999-08-01 12:02:00
Append MEMBER6 1999-08-01 12:02:00
Append MEMBER7 1999-08-01 12:02:00
Append MEMBER8 1999-08-01 12:02:00

RUED U=fc*L1 D=MEMBER3
RUED U=fc*L1 A=MEMBER2, MEMBER5, MEMBER6, MEMBER7, MEMBER8<fc*L2
RUED U=fc*L1 R=MEMBER4<fc*L2
***** END OF SYNCHRONIZE UPDATE *****

***** AUTOMATIC UPDATE DOES THE FOLLOWING: *****
Update TPR1 1999-08-15 08:15:00

RUED Update=fc*L1 Include=fc*V1
***** END OF AUTOMATIC UPDATE *****

***** AUTOMATIC UPDATE DOES THE FOLLOWING: *****
Update TPR1 1999-08-15 08:15:00

RUED Update=fc*L1 Include=fc*V1
***** END OF AUTOMATIC UPDATE *****
```

7.2 Directory Listing Report

The TABLE directive produces a member directory listing report on report code 01. The listing is sorted ascending on name.

Each directory entry contains following information:

- Name - Name of member
- Date/Time - Date and time for last update
- Size - Size of program in blocks (512W)

7.3 Error Messages

When ARUED detect any error conditions during operation, an immediate (FATAL) abort occur, thus the status of an ARUED activity may be checked as follows:

NORMAL TERMINATION - All directives processed.

ABORT-CODE "FA" - Fatal error detected during process.

7.3.1 Warning Messages

```
W-018 Waiting - Both PRIMARY and SECONDARY libraries are busy
```

Both libraries are busy, however the allocation is retried.

```
W-020 Synch overflow - complete library synch requested
```

The number of synchronization changes has exceeded ARUED limits, and a full library synchronization has been requested.

```
W-021 Delete overflow - partial synch of deleted
```

The number of synchronization deletes has exceeded ARUED limits, and only some of the deletes are synchronized.

7.3.2 Fatal Error Messages

```
F-009 Both PRIMARY and SECONDARY libraries are busy
```

ARUED cannot detect which library to update, since neither library is available for "write" allocation. Check the library files and TP8 operation.

Note: Could be caused by incorrect use of GATE resource file also.

```
F-010 Wrong file-code 'XX'
```

Internal error - report the problem.

```
F-011 I/O error occurred on filecode 'XX'
```

Permanent I/O error on filecode 'XX'. Check for disk problems on device.

```
F-013 Attempted access to sector zero on filecode 'XX'
```

Check for input file 'XX' for proper format, ex. a failed LKED activity.

```
F-015 Directive SECONDARY expected
```

SECONDARY directive must immediately follow a PRIMARY directive.

```
F-016 Directive PRIMARY expected
```

PRIMARY directive must immediately follow a GATE directive.

```
F-017 Cannot allocate filecode 'XX'
```

ARUED cannot dynamically allocate a file to filecode 'XX'. Check directive for catalog/file string and check the file for problems.

```
F-022 Allocation of XXXXXXXX failed due to YYYYYY
```

Dynamic allocation of XXXXXXXX file failed. Cause indicated in YYYYYY. Check file for problems.

```
F-031 Library format bad (not run-unit)
```

The library file has incorrect format. Check the library allocation for errors. Report problem.

```
F-032 Library format bad (sector size)
```

The library file has incorrect format. Check the library allocation for errors. Report problem.

```
F-033 Library format bad (entry count)
```

The library file has incorrect format. Check the library allocation for errors. Report problem.

```
F-033 Library format bad (control unit)
```

The library file has incorrect format. Check the library allocation for errors. Report problem.

```
F-034 File 'XX' not present
```

Internal error - report the problem.

```
F-035 Allocation of temp-file 'XX' failed
```

ARUED was unable to allocate a temp-file. Check for space problems.

F-037 Unrecognizable directive

Check input directives.

F-038 Unrecognizable argument

Check input directives.

F-039 File-code overflow (internal)

Internal error - report the problem.

Chapter 8

Software Installation/Update

This chapter describes the software installation/release update procedures and specific information related to the current release of SLFLIB (Release Bulletin).

8.1 General Description

The SLFLIB software is delivered as a set of files on a 3 1/2 inch Software Release Diskette. Alternatively the software may also be delivered as a FILSYS Software Release Tape. The installation and update procedures are quite different, and these are described separately in the following.

After installation the software must be generated with your GCOS-8 software release and then integrated into your current JCL procedures.

8.2 Software Diskette Procedure

The diskette procedure requires you are able to perform binary File-Transfer from a PC, e.g. KERMIT. Should this be a problem or if you prefer the Software Release Tape, please contact B & C Solution for a Software Release Tape.

Run a FILSYS to create the required catalogs and files. The following JCL may be used for this activity:

```
$ IDENT ....
$ FILSYS
  USERID CMDLIB$...
  CC CMDLIB/SLFLIB,R
  CC CMDLIB/SLFLIB/JCL
  CC CMDLIB/SLFLIB/SOURCE
  CC CMDLIB/SLFLIB/OBJECT
  CC CMDLIB/SLFLIB/HSTAR
  CF CMDLIB/SLFLIB/HSTAR/LODLIB,B/100,1000/,MODE/R/
```

Note: This activity is only required for initial installation.

Next perform the following file transfers using your PC terminal emulator (VIP7800, VIP3, GLINK etc.):

\SLFLIB\GATE,I -->	CMDLIB/SLFLIB/GATE
\SLFLIB\JCL\COMPFSI -->	CMDLIB/SLFLIB/JCL/COMPFSI
\SLFLIB\JCL\LINKLOD -->	CMDLIB/SLFLIB/JCL/LINKLOD
\SLFLIB\JCL\BUILDQ -->	CMDLIB/SLFLIB/JCL/BUILDQ
\SLFLIB\JCL\RUED.JCL -->	CMDLIB/SLFLIB/JCL/RUED.JCL
\SLFLIB\SOURCE\SLFGSI -->	CMDLIB/SLFLIB/SOURCE/SLFGSI
\SLFLIB\OBJECT\SLFGSI,I -->	CMDLIB/SLFLIB/OBJECT/SLFGSI
\SLFLIB\OBJECT\MODLIB,I -->	CMDLIB/SLFLIB/OBJECT/MODLIB

Note: x.x,I indicate a binary file-transfer is required.

Note: RUED.JCL must be converted to 'BCD' format.

8.3 Software Tape Procedure

Run a FILSYS user restore of the SLFLIB Software Release Tape. The following JCL may be used for this activity:

\$ IDENT
\$ USERID CMDLIB\$password
\$ FILSYS
\$ LIMITS 5,,,2K
\$ PRIVITY
RESTORE BCSTB8/SLFLIB,NEWNAM/CMDLIB/,RESET/DEVICE/
\$ TAPE9 PR,X1DD,,,99999,,,SLFLIB-SAVE,,,DEN16

Please notice PRIVITY is required to change SMC. You may select to install the software under a different SMC, however all documentation and JCL assume CMDLIB is used.

8.4 Software Customization & Generation

If you are going to use the Library Group Selection feature of SLFLIB, you need to customize the SLFGSI module with SITE specific rules.

Next you need to build the SLFLIB program file under your current GCOS-8 release.

8.4.1 Software Customization

The Library Group Selection feature is controlled through the SLFGSI module, which is located under CMDLIB/SLFLIB/SOURCE/SLFGSI. This module must be customized with SITE specific rules, otherwise the default SLFGSI module will select

ANY TPR for ANY GROUP. Following example of SITE specific rules are used to illustrate the necessary customization:

SITE have three library group's

Group A - contains all TPR's named A* or B* (* is a mask)

Group B - contains all TPR's > 100KW

Group C - contains all other TPR's

The default selection logic within SLFGSI looks as follows:

```
*- Start of customized selection logic
    perform dummy-selection.
*- End of customized selection logic
...
dummy-selection section.
section-entry.
    move TRUE to gsi-select.
section-exit.
    exit.
```

The customized selection logic within SLFGSI could look as follows:

```

*-   Start of customized selection logic

    perform site-selection.

*-   End   of customized selection logic

...

site-selection section.
section-entry.

    if ( gsi-size * 64 / 1024 ) > 100
        if gsi-group = "B"
            move TRUE to gsi-select
            go to section-exit
        else
            move FALSE to gsi-select
            go to section-exit
    else
        if gs-name-1 = "A" or
            "B"
            if gsi-group = "A"
                move TRUE to gsi-select
                go to section-exit
            else
                move FALSE to gsi-select
                go to section-exit
    else
        if gsi-group = "A" or
            "B"
            move FALSE to gsi-select
            go to section-exit
        else
            move TRUE to gsi-select
            go to section-exit.

section-exit.
exit.

```

Next you need to compile the customized SLFGSI module. The JCL for this activity is contained in the file CMDLIB/SLFLIB/JCL/COMP GSI. Insert IDENT information as required at your site.

Selection rules are applied whenever a TPR is referenced during SLFLIB execution, i.e. APPEND, REPLACE, DELETE or UPDATE. If selection rules are changed, SLFLIB will automatically DELETE a TPR from it's previous GROUP selection.

Note: If you get the rules wrong, i.e. TPR is never selected for any GROUP, then ONLY the TPR's involved in the current SLFLIB run is affected, and you may then simply adjust the rules and re-run SLFLIB to get it right.

8.4.2 Software Generation

Finally load the SLFLIB program to the System Loadable File to assure proper standard subroutines from your current GCOS-8 software release. The JCL for this activity is contained in the file CMDLIB/SLFLIB/JCL/LINKLOD. Insert IDENT information as required at your site.

The SYSOUT reports from SLFLIB and ARUED are written in BCD (report code 74) and ASCII (report code 01). This default may be changed as required. The USE statements in LINKLOD define the media codes for both reports as follows:

```
WASC01 - ASCII Report code 01
WBCD01 - BCD Report code 01
WASC74 - ASCII Report code 74
WBCD74 - BCD Report code 74
```

8.5 Software Integration

SLFLIB may be integrated into your current JCL using either of two techniques - PRMFL reference and INSERT-EDIT.

8.5.1 PRMFL Reference

This technique requires a \$ PRMFL to be inserted in each JCL procedure where SLFLIB replaces the standard LODLIB program.

```
$ IDENT ....
$ LODLIB
$ PRMFL **,R,R,CMDLIB/SLFLIB/HSTAR/LODLIB
$ PRMFL OT,W,R,....
directives
```

8.5.2 INSERT-EDIT

This technique requires no changes to JCL procedures as SLFLIB is edited into the GCOS-8 system files to replace LODLIB.

The JCL to build the Q*-tape is contained in the file CMDLIB/SLFLIB/JCL/BUILDQ. After building the Q*-tape should be edited either into your Total System Tapes or directly via Insert Edit during startup. The startup boot information should be changed to include the following control cards:

```
$EDIT SECTION
$ FILDEF dev,SLFLIB,100/0,SYS,*

$FILES SECTION
$ ETC SLFLIB,
```

The placement of the \$ ETC card should be as early in the \$FILES section as possible, to ensure override of the standard LODLIB program.

Note: Using this technique you are not able to use the standard LODLIB, thus if features like PATCH which are not supported by SLFLIB are used, this technique should not be used.

8.6 Software Release Bulletin

Following describes the changes and enhancements introduced with this version 2.3A, 2.2D, 2.2C, 2.2B, 2.2A and 2.2 of SLFLIB.

8.6.1 Warning

- The Time-Sharing CPY command does an intelligent (determines the logical size of a file) copy of files, however the calculation of logical size is not correct for LODLIB format files (does not take into account the possibility of space fragmentation).
- Use Time-Sharing COPY instead or copy the library via batch utility.
- SLFLIB will detect if such truncation has occurred (F-026 Library size too small (truncated)) and automatically adjust the space allocation, however any members outside of the current size cannot be accessed and will be dropped, when the truncated library is used as input to a new library build.

8.6.2 Problems fixed

- The size of a TPR could be incorrectly calculated for some combinations of H* layouts. This could lead to truncation of the program image (1-4096 words missing and containing low-values).
- Potential update problem for library groups in relation to Y2K has been fixed.
- Using a TP-LIB format library as input to an update of multiple library group's, would not work correctly for other than first library group. Other library groups would be update with "empty" members only.
- SLFLIB will now ensure that there is enough free space left for a TP-LIB format library. This is to ensure compatibility with standard TP-LIB, which will always allocate new space and TPR directory, regardless of what operation is done, i.e. LIST. SLFLIB will thus ensure that there is free space left for an additional space and TPR directory, so TP-LIB will not fail. Note: This would

only occur if the library size had been optimized and no room for dynamic growth was given.

- For an update operation with an H* or Q*-file build with exactly 15 members or multiple of 15 members, SLFLIB would drop the very last member from the update.
- DELETE masked, i.e. ABC*, could fail to delete all matching members, depending on their placement in the directory.
- SLFLIB would allow you to create and initialize a TP-LIB format library, even though the size was not in LINKS (multiple of 12 llinks). SLFLIB would grow the library in LINK units and TP8 would be perfectly happy with this. If, however, you use standard TP-LIB on the same library, it would abort with "SP", because the size was not in LINK units.
- SLFLIB will no longer allow you to initialize a TP-LIB format library, unless the size is in LINK units (multiple of 12 llinks). The error message "F-030 Initial library size is not in LINK units" is given.
- SLFLIB will now give an information message "I-004 The library is NOT a multiple of 12 llinks", if the current library size is not in LINK units.
- SLFLIB will now reserve space for the maximum size space directory for a TP-LIB format library (10 llinks). This is done to ensure compatibility with standard TP-LIB.
- Empty TP-LIB format libraries could not be used in TP8-mode, i.e. synchronize of empty libraries.
- DELETE of multiple TPR's did not work for TP-LIB format libraries.

8.6.3 Changes

- New TP8LIB argument for CLEAR directive, to ensure compatibility with IT4.1 release.
- SLFLIB now requires 130K for execution (previously 116K) due to the new group selection support, and the optimized I/O. Memory allocation is dynamic and has no impact on existing JCL!

8.6.4 Enhancements

- Support of Virtual TPR libraries through Advanced Run-unit Librarian - ARUED.

- Report formatting, i.e. number of lines per page, is now controlled by GCOS-8 system setting.
- SLFLIB will now set PSW switch-35 ON whenever a warning or error message is given during SLFLIB execution. Either a warning or error message indicate that the library(s) may not have been successfully updated, and switch-35 may be used to alter your JCL behavior. Note: An error message will always trigger an "FA" abort.
- SLFLIB now support dynamic library group selection.
- I/O during processing of TPR (body) has been optimized (160 versus 24 sectors in each I/O operation).

8.6.5 Migration

No special migration is required. Use of library group selection does however require change of JCL.

Chapter 9

Software Problem Reporting

9.1 General Description

The Software Problem Report procedure has been set-up to provide you, the person responsible for SLFLIB, with a consistent procedure for reporting software or documentation problems and subsequently receive replies in the form of explanations, fixes or updates of the software.

9.2 Software Problem Report

Although you may have the administration responsibility, software problems are often originating from the development system users, whether programmers, analysts or others.

The Software Problem Report (SPR) form may be used to describe the nature of such problems, any avoidance procedures taken and documentation you have collected in relation to the problems. You must fill this form for each problem and ship it to B & C Solution either through mail or facsimile. The site identification and the date of issue is the identification of the SPR. If you need assistance urgently, you may also contact our office in Denmark directly.

Once the problem has been analyzed, B & C Solution will forward a Software Problem Response (SPR) form describing the problem nature, how to avoid it, corrective actions required, an actual fix for the problem or a confirmation on when a fix will be available.

A sample SPR form is attached to this chapter and this should be copied as required.

9.3 Software Problem Distribution

Depending on the nature of a problem, B & C Solution may send the SPR forms to other SLFLIB users, in order to pre-empt a recognized problem in the software. A "Known Problem List" may also be distributed by B & C Solution, in order to document any recognized problem to all SLFLIB users.

9.4 Software Release Documentation

When new releases or versions are shipped, B & C Solution will provide you with a new manual, which will include "Software Release Bulletin" and software update information (ref. chapter 8).

Software Problem Report

Use this form to report problems, documentation errors or suggested enhancements. Please mail or fax the form to B & C Solution. The date and company name will be used as reference of SPR.

Note: For emergency problem assistance please contact B & C Solution office in Denmark directly.

Telephone: +45 49.70.47.90

Facsimile: +45 49.70.47.80

E-mail: support@bc-solution.dk

Web-site: <http://www.bc-solution.dk>

IDENTIFICATION

Date: _____

Company name: _____

Contact person: _____

Street: _____

City: _____

Country: _____

Telephone: _____

Facsimile: _____

Product Registration Number: _____

CATEGORY

Software problem

Documentation problem

Software enhancement

Other

SOFTWARE

Product _____ Rev. _____

Document _____ Rev. _____

DOCUMENTATION

Please attach listings, copies of displays, programs etc. or any other material related to reported problem.

B & C Solution use only

Date received: _____

Date resolved: _____

Action taken.....: _____/_____

_____/_____

_____/_____

PROBLEM

Please describe the problem in as much detail as possible. How to reproduce the problem, your diagnosis and corrective action or avoidance procedure taken.

Comments: