

Interactive System Productivity Facility (ISPF)



Addendum to the documentation for Software Configuration and Library Manager (SCLM) (V1R6, V1R7, V1R8)

Interactive System Productivity Facility (ISPF)



Addendum to the documentation for Software Configuration and Library Manager (SCLM) (V1R6, V1R7, V1R8)

Contents

About this document v

PTF/APAR documentation changes 1

UA38211.	1
Changes to the <i>SCLM Guide and Reference</i>	1
Changes to the <i>SCLM Messages and Codes</i>	9
UA35394, UA35395, UA35396, UA35397.	10
Summary of changes	10
Chapter 15. "Invoking the SCLM services"	11
Chapter 16. "SCLM Services"	11
UA35044, UA35045, UA5046, UA35047	13
Summary of changes	13
Chapter 9. "Using SCLM functions"	14
Chapter 15. "Invoking the SCLM services"	19
Chapter 16. "SCLM Services"	19
UA33621, UA33622, UA33623	21
Summary of changes	21

Chapter 1, "Defining the project environment".	22
Chapter 3, "Additional project manager tasks".	33
Chapter 5, "Language definition considerations"	33
Chapter 9, "Using SCLM functions"	38
Chapter 10, "Development scenario"	40
Chapter 11, "Architecture definition"	43
Chapter 13, "SCLM support for DB2".	46
Chapter 18, "SCLM macros"	47
UA36014, UA36871, UA32895	49
Chapter 9, "Using SCLM functions", subsection "Utilities (option 3)"	49
New chapter ("Leaving a member behind on promotion")	54
Chapter 15, "Invoking the SCLM services"	70
Chapter 16, "SCLM services"	70
Chapter 18, "SCLM Macros"	72

About this document

This document provides details of documentation changes associated with APARs shipped for Software Configuration and Library Manager (SCLM), Version 1 Releases 6.0, 7.0, and 8.0.

The documentation changes apply to the most recently published editions of:

- The *Software Configuration and Library Manager (SCLM) Guide and Reference*.
- The *Interactive System Productivity Facility (ISPF) Messages and Codes*.

If you have applied the relative APAR to your release of SCLM, you should read these documents in conjunction with the changes described in this Addendum.

The Addendum document lists the changes to the *SCLM Guide and Reference* and *ISPF Messages and Codes* that are required to reflect new behavior resulting from the application of APAR changes.

The changes are listed by APAR number, in reverse date order, so that the most recently released change appears at the beginning of the document. Each description shows:

- The date of the PTF release
- The APARs included in the released PTF
- Details of those APAR changes that affect documentation

Notes:

1. This document does NOT describe those APAR fixes that do not have an impact upon documentation.
2. The enhancements and corrections described in this section are only available after applying the listed APARs.

• Part 1: PTF/APAR Documentation Changes

This section lists the changes to the *SCLM Guide and Reference* and the *ISPF Messages and Codes* that are required to reflect new behavior resulting from the application of APAR changes.

The changes are listed by APAR number, in reverse date order, so that the most recently released change appears at the beginning of the document. Each description shows:

- The date of the PTF release
- The APARs included in the released PTF
- Details of those APAR changes that affect documentation

Notes:

1. This document does NOT describe those APAR fixes that do not have an impact upon documentation.
2. The enhancements and corrections described in this section are only available after applying the listed APARs.

• Part 2: General Documentation Changes

This section describes enhancements, corrections and updates in the documentation for SCLM Version 1 Releases 6.0, 7.0, 8.0, and 9.0. These changes are not associated with APAR numbers, as they do not require the application of any code updates.

The changes are listed in reverse date order. That is, the most recent documentation change appears at the beginning.

PTF/APAR documentation changes

UA38211

Release Date: 28 November 2007

The changes in this section apply to the following APARs and PTFs:

z/OS release	PTF #	APAR Abstract	APAR #	Doc Impact
1.8	UA38211 UA38217 UA38218 UA38219 UA38220	Cross-dependency database.	OA21740	Software Configuration and Library Manager (SCLM) Guide and Reference.
1.7	UA38210 UA38213 UA38214 UA38215 UA38216			Interactive System Productivity Facility (ISPF) Messages and Codes.

Changes to the *SCLM Guide and Reference*

This section details changes to the *SCLM Guide and Reference*.

Chapter 1, "Defining the project environment"

Section: "Step 6: Allocate and create the control data sets":

- Change last sentence of first paragraph to:
There are six types of VSAM data sets that can be associated with a project.
- After the description for **Secondary Audit Control**, add this new entry:

Cross-dependency data set

The Cross-dependency data set contains information used by the Where-used function.

- After the section, "Create the audit control data sets", add the following new section:

Create the Cross-dependency data set

The Cross-dependency data set is optional. It is only required if the Where-used function is to be used. Each project must have at least one cross-dependency data set for the Where-used function to be enabled. The VSAM attributes and space parameters should be the same as those used to define the accounting data set. Sample JCL for allocating and initialising the Cross-dependency data set can be found in member FLM02XDP of the ISP.SISPSAMP install library.

Section: "Step 8: Create the Project Definition": In the section, "Set the Project Control Options":

- In the list of bullet-point items (listing the things that you can use the project control options to specify), after • Audit Control data set, add the following two items:
 - Cross-dependency data set

- Cross-dependency dynamic update
- After the subsection, "Audit control data sets specification", add the following two new subsections:

Cross-dependency data set specification

The Cross-dependency data set is optional. It is only required if the Where-used function is to be used. The XDEP control option is used to specify the name of the Cross-dependency data set. The FLMALTC macro can be used to specify different Cross-dependency data sets on specific groups.

Cross-dependency data set dynamic update specification

The XDEPDYN control option is used to control the dynamic updating of the Cross-dependency data set. If XDEPDYN is set to Y, then the Cross-dependency data set is kept in sync with changes to the accounting data set. XDEPDYN may be set to N if the extra I/O activity is causing performance problems. The XDEPUPDT service should then be used to synchronise the Cross-dependency data set with the accounting data set.

Chapter 9, "Using SCLM functions"

Section: "Utilities (option 3)": In the section, "Library Utility":

- Replace Figure 50 "SCLM Library Utility - Entry Panel (FLMUS#P)" with the following:

```

Menu  _SCLM  _Utilities  _Help
-----
FLMUS#P          SCLM Library Utility - Entry Panel
Option ==>>> _____

blank Display member list      E Edit member          T Transfer owner
  A Browse account info        V View member          N NOPROM processing
  M Browse build map           C Build member         W Where used
  B Browse member              P Promote member
  D Delete member info         U Update auth code

SCLM Library:
Project . . : SLMTEST6
Group . . . DEV1
Type . . . . SOURCE
Member . . . _____ (Blank or pattern for member selection list)

Select and rank member list data . . AT (T=TEXT, A=ACCT, M=BMAP)

Enter "/" to select option
/ Hierarchy view                Process . . 3  1. Execute
/ Confirm delete                2. Submit
/ View processing options for Edit 3. View options
_ Show Member Description

F1=Help      F2=Split      F3=Exit      F7=Backward  F8=Forward  F9=Swap
F10=Actions  F11=NRETRIEV F12=Cancel

```

Figure 1. SCLM Library Utility (FLMUS#P)

- In the subsection, "Library Utility Commands", add the following new command after T:

W	Invokes the Where-used function. SCLM displays a list of components which include the selected component. Library utility commands (except W) may then be issued against the list members.
---	--

- In the subsection, "Member selection list":
 - Replace the first sentence:

"You can browse, view, delete, build, promote, or update the authorization code for members by making selections from a member selection list."

 with

"You can browse, view, delete, build, promote, display Where-used information, or update the authorization code for members by making selections from a member selection list."
 - Replace Figure 55??? "Member Selection List (FLMUSL#P)" with the following:

```

Menu  _SCLM  _Functions  _Utilities  _Test  _Help
-----
FLMUSL#P st : SLMTEST6.DEV1.SOURCE - HIERARCHY VIEW -          Member 1 of 23
Command ==>> _____ Scroll ==>> CSR

A=Account  M=Map      B=Browse   D=Delete   E=Edit     V=View
C=Build    P=Promote  U=Update   T=Transfer N=Noprom   W=WhereUsed

  Member   Status      Account  Language  Text       Chg Date  Chg Time
  ----
  AAAA          DEV1      TXT2     DEV1      2002/08/02 13:31:12
  CPYRITE      DEV1      DTL      DEV1      2002/01/21 13:08:15
  DDDD          DEV1      DTL      DEV1      2002/06/27 10:43:30
  DTL2          DEV1      DTL      DEV1      2002/01/21 13:08:04
  FLM01EQU     DEV1      HLAS     DEV1      2007/01/29 12:07:33
  FLM01MD1     DEV1      HLAS     DEV1      2002/02/14 12:24:05
  FLM01MD2     DEV1      PLIO     DEV1      2002/02/14 12:24:10
  FLM01MD3     TEST      HLAS     TEST      2002/02/14 12:23:52
  FLM01MD4     RELEASE  HLAS     RELEASE   2001/10/30 16:58:57
  FLM01MD5     RELEASE  HLAS     RELEASE   2001/10/30 16:58:57
  FLM01MD6     DEV1      HLAS     DEV1      2002/01/22 13:06:08
  HANK          DEV1      TEXT     DEV1      2002/05/24 10:26:00
  HANK2         DEV1      TEXT     DEV1      2002/04/17 11:04:40
  HANK3         DEV1      TEXT     DEV1      2002/06/27 12:57:47
  PMR60436     DEV1      COB2     DEV1      2002/01/31 12:18:10
  TESTS1       DEV1      PLIO     DEV1      2002/05/06 11:28:22
  TESTS2       DEV1      PLIO     DEV1      2002/02/15 12:01:11
  TEST6        TEST      TEXT     TEST      2002/06/17 14:11:05
  TSTPETE1     DEV1      TEXT     DEV1      2002/04/15 10:19:36
  VRCPTD1      DEV1      DTL      DEV1      2002/01/21 13:08:24
  Z1           DEV1      DTL      DEV1      2002/01/21 17:48:57
  Z2L          DEV1      DTL      DEV1      2002/01/21 17:49:14
  Z300103      DEV1      DTL      DEV1      2002/01/08 17:50:09
  ----

  F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
  F10=Actions  F11=NRETRIEV F12=Cancel

```

Figure 2. Member Selection List (FLMUSL#P)

- Replace Figure 56??? "SCLM Library Utility (FLMUS#P)" with the following:

Menu SCLM Functions Utilities Help							
FLMUSM#P st : SLMTEST6.DEV1.SOURCE - HIERARCHY VIEW -						Member 1 of 23	
Command ==>						Scroll ==> CSR	
A=Account	M=Map	B=Browse	D=Delete	E=Edit	V=View		
C=Build	P=Promote	U=Update	T=Transfer	N=Noprom	W=WhereUsed		
	Member	Status	Account	Language	Text	Chg Date	Chg Time
-	AAAA		DEV1	TXT2	DEV1	2002/08/02	13:31:12
-	CPYRITE		DEV1	DTL	DEV1	2002/01/21	13:08:15
-	copywrite copy book						
-	DDDDD				DEV1	2002/06/27	10:43:30
-	DTL2		DEV1	DTL	DEV1	2002/01/21	13:08:04
-	DTL source for panel	TTMENU					
-	FLM01EQU		DEV1	HLAS	DEV1	2007/01/29	12:07:33
-	Assembler copybook - Register	equates					
-	FLM01MD1		DEV1	HLAS	DEV1	2002/02/14	12:24:05
-	Routine Initialization						
-	FLM01MD2		DEV1	PLIO	DEV1	2002/02/14	12:24:10
-	Routine Prolog						
-	FLM01MD3		TEST	HLAS	TEST	2002/02/14	12:23:52
-	FLM01MD4		RELEASE	HLAS	RELEASE	2001/10/30	16:58:57
-	FLM01MD5		RELEASE	HLAS	RELEASE	2001/10/30	16:58:57
-	FLM01MD6		DEV1	HLAS	DEV1	2002/01/22	13:06:08
F1=Help		F2=Split		F3=Exit		F7=Backward	
F10=Actions		F12=Cancel		F8=Forward		F9=Swap	

Figure 3. Member Selection List with Hierarchy and Member Description View (FLMUSM#P)

- In the description of Status that follows Figure 56???, add another entry (after V) to the list of values for Status:

W Display Where-used Info *WHEREUSE

- After the subsection, "Transfer ownership", add the following new section:

Where-used

Enter the W line command on the SCLM Library Utility panel or on the member selection list to display a list of the components which include the selected component. A component is considered to be included if it has an entry in the Include List of another component, or if it has an *include-type* entry in the Build Map of another component. *Include-type* build map entries are: SINC, PROM, I*, PINCL*, INCL, and INCLD.

The Where-used panel shown in Figure 4 on page 5 displays the list of components which include the selected component.

```

Menu  SCLM  Functions  Utilities  Test  Help
-----
FLMUSF#P d : LEEBURR.DEV1.SOURCE(FLM01EQU)                               Member 1 of 38
Command ==> _____ SCROLL ==> CSR

A=Account  M=Map      B=Browse  D=Delete  E=Edit    V=View    Dyn(ON)
C=Build    P=Promote  U=Update  S=UOW/Edit  T=Transfer  N=NOPROM  X=Expand
-----1-----
S Member          Lv Type    AcctGrp  BmapGrp  AcctLang  BmapLang  Keyword
- ASM@N001        1 SOURCE  DEV1     DEV1     NOBJ      GLMAP     SINC
- ASM@N002        1 SOURCE  DEV1     *NOBMAP* NOBJ      *NOBMAP*  SINC
- + FLM01CMD       1 ARCHDEF  DEV1     DEV1     ARCHDEF   CCMAP     I1*
- + FLM01MD3       1 SOURCE  DEV1     DEV1     HLAS      GLMAP     I1*
- + FLM01MD4       1 SOURCE  RELEASE  DEV1     HLAS      GLMAP     I1*
- + FLM01MD5       1 SOURCE  RELEASE  DEV1     HLAS      GLMAP     I1*
- + FLM01MD6       1 SOURCE  RELEASE  DEV1     HLAS      GLMAP     I1*
- + LB@SRC01       1 SOURCE  DEV1     DEV1     HLAS      CCMAP     SINC
- NEW@MD01        1 SOURCE  DEV1     *NOBMAP* NOBJ      *NOBMAP*  SINC
- NEW@MD02        1 SOURCE  DEV1     DEV1     HLAS      CCMAP     SINC
- + TST@SCLM      1 SOURCE  TEST     TEST     C20B000  GLMAP     I1*
***** Bottom of data *****

F1=Help      F2=Split    F3=Exit     F7=Backward  F8=Forward  F9=Swap
F10=LEFT    F11=RIGHT   F12=Cancel

```

Figure 4. Where-used panel (FLMUSF#P)

The fields that appear on the Where-used panel are:

S	<p>Input field for row command.</p> <p>All row commands described in "Library Utility commands", except for the W command, can be used. Two additional commands are available:</p> <p>X Can be used to expand and collapse nested rows (where possible).</p> <p>S Invokes Unit Of Work if the selected member is an Archdef; otherwise, Edit is invoked.</p> <p>Note: You must exit and re-enter the panel to refresh the WhereUsed list when parts are added or deleted by commands issued from the WhereUsed member list.</p>
Name	<p>The name of the including (or parent) components. The names are indented according to nesting level and the field may be scrolled left and right. The following indicators can appear to the left of the name:</p> <p>+ The row can be expanded.</p> <p>- The row can be collapsed.</p> <p>* The row has a circular reference and will not be expanded.</p> <p>< The row has exceeded the nesting level.</p>
Lv	The nesting level.
Type	The type of the member.
AcctGrp	<p>Group where the member's account record was found.</p> <p>If the Acct record is not found, then AcctGrp shows "*NOACCT*".</p>

BmapGrp	Group where the member's build map was found. If the Bmap record is not found, then BmapGrp shows <i>"*NOBMAP*"</i> .
AcctLang	Language of the member's account record.
BmapLang	Language of the member's build map record.
Keyword	The keyword used when referring to the included component.

Chapter 15, "Invoking the SCLM services"

Section: "SCLM service descriptions": Add the following two items to the services listed:

- GETXDEP—Return cross-dependency information
(Add after GETBLDMP—Retrieve Build Map Information)
- XDEPUPDT—update the Cross-dependency data set
(Add after VERRECOV—Recover a Version)

GETXDEP—return cross-dependency information: (Add this new section after the section, "GETBLDMP—Retrieve Build Map Information".)

The GETXDEP service returns information from the Cross-dependency database associated with the specified SCLM-managed member. The information is returned in an ISPF table. One row is returned for each child-parent dependency found.

Command invocation format:

```
FLMCMD GETXDEP,project,
prj_def ,
group ,
type ,
member ,
xdep_table ,
scope,
[dd_msgs]
```

Call invocation format:

```
Lastrc := FLMLNK('GETXDEP',sclm_id,
group ,
type ,
member ,
xdep_table ,
scope,
$msg_array);
```

Parameters:

project	The project name.
prj_def	The project definition name.
group	The group associated with the accounting record.
type	The type associated with the accounting record.
member	The member under SCLM control.
xdep_table	The name of the ISPF table to contain the cross-dependency entries. The table must be open before calling the GETXDEP service. A TBADD is performed for each cross-dependency record found. The following ISPF variables must be defined to the table in order to have their value stored in the table:

ZSFGROUP	Initial group (same on all records).
ZSFSCOPE	Scope (same on all records).
ZSFLEVEL	Nesting level.
ZSFCNAME ^{Note 1}	Child object - name.
ZSFCPGRP ^{Note 1}	Child object - group.
ZSFCTYPE ^{Note 1}	Child object - type.
ZSFCLANG ^{Note 1}	Child object - language.
ZSFCPBGPR ^{Note 1}	Child object - group (buildmap).
ZSFCBLNG ^{Note 1}	Child object - language (buildmap).
ZSFCNUMP ^{Note 1}	Child object - number of parents.
ZSFPKWRD	Parent object - include keyword.
ZSFPNAME	Parent object - name.
ZSFPTYPE	Parent object - account type.
ZSFPGRP	Parent object - account group.
ZSFPBGRP	Parent object - buildmap group.
ZSFPLANG	Parent object - account language.
ZSFPBLNG	Parent object - buildmap language.
ZSFPNUMP	Parent object - number of parents.
ZSFLIMIT	Limited list = blank, "CIRC", or "MAX".

Notes:

1. The ZSFC** fields are repeated for each of their parent rows.

scope

Valid values are:

NEXT	Just return immediate parents.
ALL	Return all parents.
LMOD	Stop at LECMAP archdefs.
TOPSRC	Stop at CCMAP archdefs.

dd_msgs

The ddname indicating the destination of the messages generated by the GETXDEP service. If you specify a blank ddname, SCLM routes the GETXDEP messages to the default output device, such as your terminal. Otherwise, before you call the GETXDEP service, you must allocate the ddname. The following attributes should be used: RECFM=F, LRECL=80, BLKSIZE=80. The maximum parameter length is 8 characters. This parameter is used for FLMCMD only.

\$msg_array

An output parameter pointing to the message array. See the section "Pointer Parameter Descriptions" in Chapter 15, "Invoking the SCLM services" for more information about \$msg_array. This parameter is used for FLMLNK only.

Return codes:

0	GETXDEP completed successfully.
4	The parent chain was truncated when the maximum nesting level was exceeded, or a circular reference was detected. ZSFLIMIT is non-blank on truncated rows.
8	No parent data was found.
12	Cross-dependency database is not active.
16	Error updating ISPF table.

20 Error reading the Cross-dependency database.

XDEPUPDT—Update Cross-dependency Information: (Add this new section after the section, "VERRECOV—Recover a Version".)

The XDEPUPDT service reads the SCLM Account files and updates the associated data in the cross-dependency files.

Command invocation format:

```
FLMCMD XDEPUPDT,project,  
prj_def ,  
*|group ,  
[dd_xdeprept]
```

Call invocation format:

```
Lastrc := FLMLNK('XDEPUPDT' , sclm_id  
*|group ,  
$msg_array);
```

Parameters:

- | | |
|--------------------|--|
| project | The project name. |
| prj_def | The project definition name. |
| group | The group pattern to process. This should normally be specified as '*' to process ALL groups. |
| dd_xdeprept | The ddname indicating the destination of the messages generated by the XDEPUPDT service. If you specify a blank ddname, SCLM routes the XDEPUPDT messages to the default output device, such as your terminal. Otherwise, before you call the XDEPUPDT service, you must allocate the ddname. The following attributes should be used: RECFM= F, LRECL=80, BLKSIZE=80. The maximum parameter length is 8 characters. This parameter is used for FLMCMD only. |
| \$msg_array | An output parameter pointing to the message array. See the section "Pointer Parameter Descriptions" in Chapter 15, "Invoking the SCLM services" for more information about \$msg_array. This parameter is used for FLMLNK only. |

Chapter 18, "SCLM macros"

FLMCNTRL macro:

- In the subsection, "Macro format", add the following new parameters:
[,XDEP=xdep_data_set|project.XDEP.FILE]
[,XDEPDYN=Y|N]
- In the subsection, "Parameters", add the following new parameter descriptions:
XDEP=xdep_data_set|project.XDEP.FILE
The name of the cross-dependency file for the project. The name you specify must be the name of the VSAM cluster you want to use. The default cross-dependency file name is *project.XDEP.FILE*, where *project* is the name specified on the FLMABEG macro. The XDEP parameter on the FLMALTC macro can be used to override the cross-dependency file name for a specific group or set of groups. No SCLM variables can be used for this parameter.

Note: The XDEP data set name supplied on the FLMCNTRL macro is used if the XDEP parameter is not specified on any FLMALTC macros. If one FLMALTC macro supplies an XDEP data set name, then all FLMALTC macros must supply an XDEP data set name if no XDEP value is specified on the FLMCNTRL macro.

XDEPDYN=Y|N

Indicates whether the Cross-dependency data set is to kept in sync with the accounting data set. If Y is specified, then when each accounting data set record is updated the changes are analysed and one or more cross-dependency records may be updated. Maintaining the Cross-dependency database dynamically requires additional I/O operations. Specify N if the extra I/O activity creates performance problems. If N is specified, then it is necessary to run the XDEPUPDT service to synchronise the cross-dependency data from the accounting data set.

FLMALTC macro:

- In the subsection, "Macro format", add the following new parameter:
[,XDEP=xdep_data_set]
- In the subsection, "Parameters", add the following new parameter description:

XDEP=xdep_data_set

The name of the cross-dependency file to be used to by any group referencing this FLMALTC macro. The name you specify must be the name of the VSAM cluster you want to use. No SCLM variables can be used for this parameter.

If one FLMALTC macro supplies an XDEP data set name, then all FLMALTC macros must supply an XDEP data set name if an XDEP data set name is not specified on the FLMCNTRL macro.

Changes to the *SCLM Messages and Codes*

This section details changes to the *SCLM Messages and Codes*.

Chapter 3, "SCLM messages"

Add the following new messages:

FLM398Z	No dependency info - Dependency record not found.	FLM399D	List was truncated - The maximum nesting level was reached. Entries indicated by a < sign have parents but they will not be displayed.
Explanation:	This message is self explanatory.	Explanation:	This message is self explanatory.
FLM399A	Dependency Inactive - No dependency database is defined for this project.	FLM399E	Invalid selection.
Explanation:	This message is self explanatory.	Explanation:	This message is self explanatory.
FLM399B	Database error - An error occurred accessing the Cross Dependency database.	FLM399F	Circular reference - This object will not be expanded since it refers to itself either directly or indirectly.
Explanation:	This message is self explanatory.	Explanation:	This message is self explanatory.
FLM399C	No subentries - Row can not be expanded or collapsed.	FLM85500	ERROR RC AAAAAA READING ACCOUNT RECORD FORBBBBB
Explanation:	This message is self explanatory.		

Explanation: The XDEPUPDT utility could not read an Account record. Processing is terminated.

**FLM85501 ERROR RC AAAAAA READING
BUILDMAP RECORD FOR BBBBBB**

Explanation: The XDEPUPDT utility could not read a Buildmap record. Processing is terminated.

**FLM85505 NO CROSS DEPENDENCY ENTRIES
SELECTED**

Explanation: The XDEPUPDT service did not find any objects with dependencies. No records will be written to the database.

**FLM85506 ERROR WRITING CROSS
DEPENDENCY RECORD KEY:
AAAAAAA BBBBBBBB CCCCCC
CODE: EEEEEEE**

Explanation: The XDEPUPDT utility experienced an error writing a database record.

User response: Check for preceding error messages.

FLM85508 PROCESSING COMPLETED

Explanation: The XDEPUPDT service completed successfully.

**FLM85509 PROCESSING TERMINATED WITH
ERRORS**

Explanation: The XDEPUPDT service did not complete successfully.

**FLM85514 NO CROSS DEPENDENCY
INFORMATION FOUND FOR
GROUP:AAAAAAA TYPE:
BBBBBBB MEMBER:CCCCCCC**

Explanation: There was no record in the Cross-dependency database for this member.

**FLM85515 NO CROSS DEPENDENCY DATABASE
ACTIVE FOR PROJECT: AAAAAAA**

Explanation: The project has not defined a cross dependency database.

UA35394, UA35395, UA35396, UA35397

Release Date: 3 August 2007

The changes in this section apply to the following APARs and PTFs:

z/OS release	PTF #	APAR Abstract	APAR #	Doc Impact
1.8	UA35396 UA35406 UA35407 UA35408 UA35409	Provide additional parameters to the GETBLDMP service	OA20514	V1R8 Software Configuration and Library Manager (SCLM) Guide and Reference (SC34-4817-06).
1.7	UA35395 UA35402 UA35403 UA35404 UA35405			
1.6	UA35394 UA35398 UA35399 UA35400 UA35401			

Summary of changes

Changes relating to the PTF for SCLM APAR OA20514

Provide additional parameters to the GETBLDMP service to return additional information.

Chapter 15. "Invoking the SCLM services"

ISPF Variables

In the table containing the ISPF variables add these before the ZSBKWRD entry:

Variable	Max Size	Services	Description
ZSBMGRP	8	GETBLDMP	Build map group name
ZSBMTYP	8	GETBLDMP	Build map type name
ZSBMMEM	8	GETBLDMP	Build map member name

In addition, add this variable after the ZSBKWRD entry:

Variable	Max Size	Services	Description
ZSBGRP	8	GETBLDMP	Build map entry group name

Chapter 16. "SCLM Services"

GETBLDMP

- ISPF interface panel

The services panel should be changed to the panel shown here.

```
Menu  SCLM  Utilities  Options  Help
-----
SCLM FLMCMD GETBLDMP Service - Entry Panel
Command ===>
SCLM Library:
Project . . . ADMIN07
Alternate . .
Group . . . . DEV1
Type . . . . COBOL
Member . . .
DD Name for output data set:
Error message data set          (Blank to write messages to the terminal)
```

Figure 5.

- Parameters

The following changes need to be made to the member and bmap_table parameters.

member

The member under SCLM control. The maximum parameter length is 8 characters. You can specify an individual member, an asterisk for all of them, or a valid pattern.

bmap_table

The name of the ISPF table to contain the build map entries. The table must be open before calling the GETBLDMP service. A TBADD will be performed for each build map record.

The following ISPF variables must be used in the table definition in order to have their value stored in the table. All of the variables are optional and only the variables required need to be specified:

ZSBMGRP	Group name where build map was located
ZSBMTYP	The type associated with the accounting record for which the build map was requested. This will be the same as was specified in the <i>type</i> parameter.
ZSBMMEM	Member name(s) of build map entries that match the pattern specified in the <i>member</i> parameter.
ZSBKWRD	entry type. For example the keyword used in the ARCHDEF or language definition, such as SINC, OBJ, and LOAD.
ZSBGRP	Build map entry group name. Group where the account record for the build map entry is stored in SCLM. Note : If this parameter is specified in the table then the account records for each build map entry will be searched to locate the group where the account record exists. If a wildcard is used for member, or the build map is large, this call may take some time.
ZSBMEM	member name
ZSBTYPE	member type
ZSBDATE	build date (in YYYYMMDD format)
ZSBTIME	build time (in HHMMSS format)
ZSBVER	version
ZSBLINE	unformatted data line (72 characters)

The maximum parameter length is 8 characters (except for ZSBLINE).

Examples

Add a new sub topic for examples with the following text:

This sample REXX code is an example of how to invoke the GETBLDMP service.

```

/* REXX */

address ispexec
'CONTROL ERRORS RETURN'
'TBEND  BMAPTABL'
'CONTROL ERRORS CANCEL'
tbnames = 'NAMES(ZSBMGRP ZSBMTYP ZSBMMEM ZSBKWRD ZSBMEM ZSBTYPE
          ZSBDATE ZSBTIME ZSBVER  ZSBLINE ZSBGRP)'
'TBCREATE BMAPTABL' tbnames 'NOWRITE'
'TBVCLEAR BMAPTABL'

parmbmap = 'ADMIN10,ADMIN10,DEV1,COBOL,RDBK*'
Address TSO 'FLMCMD GETBLDMP,||parmbmap||',BMAPTABL'
Say 'Return code from Build Map lookup = 'rc
If RC < 5 Then
Do
  'TBTOP  BMAPTABL'
  'TBSKIP BMAPTABL'

Say ''
Say 'BM Group BM Type  BM Memb ' ,
  'Keyword Member  Type    Group   Date/Time Modified  Version'
Do While (RC = 0)
  Say SUBSTR(ZSBMGRP,1,8) SUBSTR(ZSBMTYP,1,8) SUBSTR(ZSBMMEM,1,8) ,
    SUBSTR(ZSBKWRD,1,8) SUBSTR(ZSBMEM,1,8) SUBSTR(ZSBTYPE,1,8),
    SUBSTR(ZSBGRP,1,8) ZSBDATE' 'ZSBTIME ' ' ZSBVER ' '

```

```

        'TBSKIP BMAPTABL'
    End
    'TBEND BMAPTABL'
End

```

This example builds the temporary ISPF table BMAPTABL with all of the available parameters. The service call is invoked for COBOL members beginning with RDBK, starting at the DEV1 group in the hierarchy. Once complete the temporary ISPF table is read to list out the build map entries.

The output will be as follows: <

Return code from Build Map lookup = 0

BM Group	BM Type	BM Memb	Keyword	Member	Type	Group	Date/Time	Modified	Version
DEV1	COBOL	RDBKC02	SINC	RDBKC02	COBOL	DEV1	20070625	155226	1
DEV1	COBOL	RDBKC02	OBJ	RDBKC02	OBJ	DEV1	20070625	155357	1
DEV1	COBOL	RDBKC02	LIST	RDBKC02	LIST	DEV1	20070625	155357	1
DEV1	COBOL	RDBKC02	OUT1	RDBKC02	DBRM	DEV1	20070625	155357	1
DEV1	COBOL	RDBKC02	I1*	ORDRSET	COPYBOOK	TEST	20070607	114654	1
DEV1	COBOL	RDBKC02	I1*	RDBKT01	COPYBOOK	TEST	20070522	165926	1
TEST	COBOL	RDBKC01	SINC	RDBKC01	COBOL	DEV1	20070612	143325	3
TEST	COBOL	RDBKC01	OBJ	RDBKC01	OBJ	TEST	20070625	155347	14
TEST	COBOL	RDBKC01	LIST	RDBKC01	LIST	TEST	20070625	155347	14
TEST	COBOL	RDBKC01	OUT1	RDBKC01	DBRM	TEST	20070625	155347	13
TEST	COBOL	RDBKC01	I1*	ORDRSET	COPYBOOK	TEST	20070607	114654	1
TEST	COBOL	RDBKC01	I1*	RDBKT01	COPYBOOK	TEST	20070522	165926	1
TEST	COBOL	RDBKC01	I1*	RDBKDC01	DCLGEN	TEST	20070612	143307	1

UA35044, UA35045, UA5046, UA35047

Release Date: 3 July 2007

The changes in this section apply to the following APARs and PTFs:

z/OS release	PTF #	APAR Abstract	APAR #	Doc Impact
1.8	UA35046 UA35056 UA35057 UA35058 UA35059			
1.7	UA35045 UA35052 UA35053 UA35054 UA35055	Provide a new mode that allows the build to be run in an information mode.	OA21104	V1R8 Software Configuration and Library Manager (SCLM) Guide and Reference (SC34-4817-06).
1.6	UA35044 UA35048 UA35049 UA35050 UA35051			

Summary of changes

Changes relating to the PTF for SCLM APAR OA21104

Provide a new mode that allows the build to be run in an information mode. This mode will return in XML format information relating to the build translator, such as the build options and the data set concatenations used for include dependencies.

Chapter 9. "Using SCLM functions"

Build (option 4) section

The paragraph that reads "You also can specify that a Build Report be generated without actually invoking any translators. The Build Report identifies those components in the hierarchy that would change if translators were to be invoked", should have a new paragraph added after it with the following text:

A build can also be run in information mode. When this is done, the options passed to the translators, and the data sets used in the concatenations for resolving the include dependencies, are written to the build report output in XML format. The options that would be passed to a compiler will be shown, after FLMTOPTS and Compilation Control architecture member overrides have been taken into consideration. In addition, all DDs in the SYSLIB, or any other IOTYPE=I allocation, will be shown as they would be passed to the compiler with all FLMCPYLB, FLMINCLS and FLMSYSLB overrides taken into consideration.

The panel shown in Figure 5 on page 11 appears when you select Option 4, Build, from the SCLM Main Menu.

```
Menu  SCLM  Utilities  Jobcard  Test  Workstation  Build  Help
-----
SCLM Build - Entry Panel
Command ==>
Build input:
Project . . : PDF42SVT
Group . . . DEVI
Type . . . . ARCHDEF
Member . . . SAMPLE
Enter "/" to select option
/ Error Listings only
  Workstation Build
Mode . . . 1 1. Conditional
                2. Unconditional
                3. Forced
                4. Report
                5. Information
Scope . . . 2 1. Limited
                2. Normal
                3. Subunit
                4. Extended
Output control:
Ex Sub
Messages . . 3 3 1. Terminal
Report . . . 3 3 2. Printer
Listings . . 3 3 3. Data set
                4. None
Process . . . 1 1. Execute
                2. Submit
Printer . . . H
Volume . . . _____
```

Figure 6. SCLM Build (FLMB#P)

The table that describes the fields on the screen needs to have a new mode added. After mode "Report", add the following new element to the table.

Information

The information Build returns an XML document describing the input that would have been passed to the BUILD translators on a Forced Build. The information is returned in the build report and contains the options passed to the translators, along with the data set allocations made by SCLM for IOTYPE=I allocations, which are include dependency allocations.

After the Build report Example section add a new section:

Build Information Example

Build information is returned in the build report data set. Just like the normal report mode build, the title page identifies the date and time of the

build, as well as the scope and mode used. It also lists the member you specified on the Build panel and the project definition specified on the SCLM Main Menu.

The report lists, in XML format, all members in the scope of the build that would have been built if the build had been performed in "Forced" mode. For each member, the language is listed and each translator within the language definition is shown. For each translator, the options passed to the translator, along with any include dependency DD allocations, are shown.

Figure 7 on page 16 shows an example of a build information report.

```

*****
*****
**
**
**          SOFTWARE CONFIGURATION AND LIBRARY MANAGER (SCLM)
**
**          B U I L D   R E P O R T
**
**          2007/06/18   16:33:02
**
**          PROJECT:     ADMIN10
**          GROUP:       DEV1
**          TYPE:        COBOL
**          MEMBER:      RDBKC01
**          ALTERNATE:   ADMIN10
**          SCOPE:       NORMAL
**          MODE:        INFORMATION
**
**
*****
*****

```

```

<?xml version="1.0"?>
<build-information>
  <member name="RDBKC01" lang="COBOL"
    langdesc="ENTERPRISE COBOL WITH CICS AND DB2">
    <translator name="IGYCRCTL">
      <option-set>
        <option-string>NONUM,LIB,XREF(FULL),MAP,OFFSET,</option-string>
        <option-string>NOOPTIMIZE,CICS(&quot;COBOL&quot;),SQL</option-string>
      </option-set>
      <data-definition name="SYSLIB">
        <dsn name="ADMIN10.DEV1.COBOL"/>
        <dsn name="ADMIN10.TEST.COBOL"/>
        <dsn name="ADMIN10.PROD.COBOL"/>
        <dsn name="ADMIN10.DEV1.COPYBOOK"/>
        <dsn name="ADMIN10.TEST.COPYBOOK"/>
        <dsn name="ADMIN10.PROD.COPYBOOK"/>
        <dsn name="ADMIN10.DEV1.DCLGEN"/>
        <dsn name="ADMIN10.TEST.DCLGEN"/>
        <dsn name="ADMIN10.PROD.DCLGEN"/>
        <dsn name="CICS.TS31.CICS.SDFHCOB"/>
        <dsn name="CICS.TS31.CICS.SDFHMAC"/>
      </data-definition>
      <data-definition name="DCLGEN">
        <dsn name="ADMIN10.DEV1.DCLGEN"/>
        <dsn name="ADMIN10.TEST.DCLGEN"/>
        <dsn name="ADMIN10.PROD.DCLGEN"/>
      </data-definition>
    </translator>
  </member>
</build-information>

```

Figure 7. Build Information Report

Note: Standard XML rules apply to certain special characters, called pre-defined entities. If there are any of the pre-defined entities in the translator options they will be converted to the required XML format name as specified in Table 1

Table 1. Pre-defined entities

Character value	Pre-defined entity value	Description
<	<	left angle bracket

Table 1. Pre-defined entities (continued)

Character value	Pre-defined entity value	Description
>	>	right angle bracket
&	&	ampersand
'	'	single quote character
"	"	double quote character

Description of the Elements and Attributes in the Build information XML

The following table contains a description of the elements and attributes that make up the build information XML document.

Table 2. Description of the Elements and Attributes in the Build information XML

Name	Description
build-information	<p>Root element. There is a single <i>build information</i> element.</p> <p>Sub-elements: member</p> <p>Attributes: None</p>
member	<p>There is one <i>member</i> element for each member in the scope of the build information report. For example; if build information is requested on an LEC architecture definition, then there will be a <i>member</i> element for each member in the LEC architecture definition.</p> <p>Sub-elements: translator</p> <p>Attributes: name Name of the member lang SCLM language of the member langdesc SCLM language description</p>
translator	<p>There is one <i>translator</i> element for each translator step in the SCLM language definition. For example; if a COBOL compile has a separate DB2 pre-compile, then there will be a <i>translator</i> element for the DB2 pre-compile and a <i>translator</i> element for the COBOL compile step.</p> <p>Sub-elements: option-set data-definition</p> <p>Attributes: name Name of the translator invoked. For example; IGYCRCTL for the COBOL compiler</p>
option-set	<p>There is one <i>option-set</i> element for each translator. This will contain a number of <i>option-string</i> elements depending on how many options there are.</p> <p>Sub-elements: option-string</p> <p>Attributes: None</p>

Table 2. Description of the Elements and Attributes in the Build information XML (continued)

Name	Description
option-string	<p>There are one or more <i>option-string</i> elements depending on the number of options specified in the translator. This element contains as many options as will fit within an 80 byte record.</p> <p>Sub-elements: None</p> <p>Attributes: None</p>
data definition	<p>There is one <i>data-definition</i> element for each IOTYPE=I (Include dependency) allocation in the translator. This will contain a number of <i>dsn</i> elements depending on how many data sets are allocated to the specified DD.</p> <p>Sub-elements: dsn</p> <p>Attributes: name Name of the DD specified in the FLMALLOC macro for an IOTYPE=I allocation. For example; SYSLIB and DCLGEN.</p>
dsn	<p>There is one <i>dsn</i> element for each data set that is allocated as part of the concatenation of DDs for the DDNAME specified in the <i>name</i> attribute of the <i>data-definition</i> element.</p> <p>Sub-elements: None</p> <p>Attributes: name Data set name</p>

Chapter 15. "Invoking the SCLM services"

ISPF Variables

In the table containing the ISPF variables add the following:

ZSCIAUT	32KB	SCLMINFO	Authorization code information for the project
ZSCINPAT	32KB	SCLMINFO	Data set name patterns information for the project

Chapter 16. "SCLM Services"

Build

Command invocation format

Add the mode of I to the list of valid modes.

```
FLMCMDBUILD,project
, [prj_def]
, group
, type
, member
, [userid]
, [E|L|N|S]
, [C|F|R|U|I]
, [Y|N]
, [Y|N]
, [prefix_userid]
, [dd_bldmsgs]
, [dd_bldrept]
, [dd_bldlist]
, [dd_bldexit]
```

Call invocation format

Add the mode of I to the list of valid modes.

```
lastrc := FLMLNK('BUILD ',sclm_id
,group
,type
,member
,{userid|' '}
,{E|L|N|S}
,{C|F|R|U|I}
,{Y|N}
,{Y|N}
,[{prefix_userid|' '}
,[dd_bldmsgs
,[dd_bldrept
,[dd_bldlist
,[dd_bldexit]]]]);
```

ISPF interface panel

The services panel should be changed to the sample shown below:

```

Menu  SCLM  Utilities  Options  Help
-----
SCLM FLMCMD BUILD Service - Entry Panel
Command ==>>

SCLM Library:
Project . . . SCLMTEST
Alternate . .
Group . . . . DEV1
Type . . . . SOURCE
Member . . .

Enter "/" to select option
/ Error listings only
_ Create build report

Mode . . 1 1. Conditional      Scope . . 2 1. Limited
          2. Unconditional    2. Normal
          3. Forced           3. Subunit
          4. Report           4. Extended
          5. Information

User id . . . . . DOHERTL (If blank, your user id is used)
Prefix for temporary data sets _____ (Blank to default to user id)

DD Names for output data sets:
Messages . . _____ (Blank to write messages to the terminal)
Report . . . _____ (Blank to write report to the terminal)
Listings . . _____ (Blank to write list to the terminal)
Exit . . . . _____ (Blank if no exit or no output required)

```

Figure 8. BUILD Service panel

Parameters

The mode parameters need to be changed to reflect the new parameter as shown below:

C|F|R|U|I

Indicates the build mode (C=conditional, F=forced, R=report, U=unconditional, I=Information) For FLMCMD, the default is C. There is no default for FLMLNK. The maximum parameter length is 24 characters.

SCLMINFO

The following variables need to be added to the list of ISPF variables returned by the service:

ZSCIAUT Authorization code information for the project
ZSCINPAT Data set name patterns information for the project

In the descriptions of the variables after the list the following descriptions need to be added:

ZSCIAUT contains a list of all the authorization codes specified at each group for this project. Here is an example of the information that is returned for each group:

GR=TEST AC=P D

You must parse the variable to retrieve the authorization codes for each group.

ZSCINPAT contains a list of all the data set patterns used for each group for this project. Even if flexible data set naming has not been used on the FLMCNTRL or FLMLTTC macros this variable will be returned with the default pattern of

@@FLMPRJ.@@FLMGRP.@@FLMTYP specified for each group. Here is an example of the information that is returned for each group:

GR=PROD NP=@@FLMPRJ.V2.@@FLMGRP.@@FLMTYP

You must parse the variable to retrieve the data set pattern for each group.

UA33621, UA33622, UA33623

Release Date: 20 April 2007

The changes in this section apply to the following APARs and PTFs:

z/OS release	PTF #	APAR Abstract	APAR #	Doc Impact
1.8	UA33623 UA33632 UA33633 UA33634 UA33635			
1.7	UA33622 UA33628 UA33629 UA33630 UA33631	Improve sample application to use updated compilers and extend to use COBOL sample.	OA17586 (and also OA20500)	V1R8 Software Configuration and Library Manager (SCLM) Guide and Reference (SC34-4817-06).
1.6	UA33621 UA33624 UA33625 UA33626 UA33627			

Summary of changes

Changes relating to the PTF for SCLM APAR OA17586

- The first optional compiler in the SCLM Sample Project is changed from Optimizer PL/I to Enterprise PL/I.
- Enterprise COBOL is added to the SCLM Sample Project as the second optional compiler.
- Optional Fault Analyzer is added to the SCLM Sample Project.
- A Language Definition and sample program are added to the SCLM Sample Project to demonstrate Fault Analyzer side file generation and Assembler compile options override on the CC architecture member.
- SCLM option 10.7.1 creates a JCL member which can be used to re-compile the SCLM Sample Project Definition.
- Made consistent use of FLMTYP LIST in:
 - Enterprise compiler language definitions
 - All other language definitions which are included in the SCLM Sample Project
 - All listings in this documentation
- Six new dialog variables are added to the SCLM Sample Project:
 - ZQCOBE(select Enterprise COBOL)
 - ZQCDSN(Enterprise COBOL Compiler location)
 - ZQPLIE(select Enterprise PL/I)
 - ZQCSYS1(COBOL SYSLIB Library)

- ZQFAAV(select Fault Analyzer)
 - ZQIDSN(Fault Analyzer location)
- ZQPLIO and ZQPSYS2 are no longer in use.

Chapter 1, "Defining the project environment"

"Step 5: Allocate the project partitioned data sets" section

Table 3 (Data Set Attributes)

Replace:

Link-edit Maps	FBM	121
----------------	-----	-----

with:

Link-edit Maps	FBA	121
----------------	-----	-----

"Step 8: Create the project definition" section

Figure 15 (Sample Alternate Project Definition)

Replace:

```
COPY FLM@COBL -- COBOL LANGUAGE --
```

with:

```
COPY FLM@COBE -- Enterprise COBOL LANGUAGE --
```

Replace:

```
COPY FLM@PLIO -- PL/I OPTIMIZER LANGUAGE --
```

with:

```
COPY FLM@PLIE -- Enterprise PL/I LANGUAGE --
```

Figure 16 (Example Project Definition)

After:

```
DOCS FLMTYPE
```

add:

```
IDILANGX FLMTYPE
```

After:

```
COPY FLM@COB2 -- COBOL II    LANGUAGE --
```

add:

```
COPY FLM@COBE -- Enterprise COBOL LANGUAGE --
```

After:

```
COPY FLM@FORT -- FORTRAN IV LANGUAGE --
```

add:

```
COPY FLM@HLAF -- HIGH LEVEL ASSEM. LANGUAGE --
*              -- WITH FAULT ANALYZER      --
```

After:

```
COPY FLM@PLIO -- PL/I OPTIMIZER LANGUAGE --
```

add:
 COPY FLM@PLIE -- Enterprise PL/I LANGUAGE --

Replace:
 * 5665-402 (C) COPYRIGHT IBM CORP 1992, 1990

with:
 * 5694-A01 COPYRIGHT IBM CORP 1992, 2007

Table 4 (Language Definitions Supplied with SCLM)

After:

COBOL II with member expansion and CICS preprocessing	FLM@ICO2
---	----------

add:

Enterprise COBOL	FLM@COBE
COBOL with integral CICS preprocessing	FLM@CCBE
Enterprise COBOL with integral DB2 preprocessing	FLM@2CBE
Enterprise COBOL with integral DB2 and CICS preprocessing	FLM@2CCE

After:

PL/I Optimizer with member expansion and CICS preprocessing	FLM@IPLO
---	----------

add:

Enterprise PL/I	FLM@PLIE
Enterprise PL/I with integral CICS preprocessing	FLM@CPLIE
Enterprise PL/I with integral DB2 preprocessing	FLM@2PLIE

"Modifying example language definitions" subsection

- Replace the sentence before Figure 17 (OS/VS COBOL Language Definition Example):
 "Figure 17 on page 39 shows an example of an OS/VS COBOL language definition."

with:

"Figure 17 on page 39 shows an example of an Enterprise COBOL language definition."

- Replace Figure 17 (OS/VS COBOL Language Definition Example) with the following:

```
*****
*           ENTERPRISE COBOL LANGUAGE DEFINITION FOR SCLM           *
*                                                                 *
*****
*
*           FLMLANGL    LANG=COBE,                                  C
*           LANGDESC='ENTERPRISE COBOL ',                          C
*
*
```

```

*****
*          --PARSER TRANSLATOR--          *
*****
*
*          FLMTRNSL  CALLNAM='SCLM COBOL PARSE',          C
*                   FUNCTN=PARSE,                      C
*                   COMPILE=FLMLPCBL,                  C
*                   PORDER=1,                          C
*                   CALLMETH=LINK,                     C
*                   OPTIONS=(@@FLMLIS,@@FLMSTP,@@FLMSIZ,)
*          (* SOURCE *)
*          FLMALLOC  IOTYPE=A,DDNAME=SOURCE
*          FLMCPYLB  @@FLMDSN(@@FLMMBR)
*
*****
*          --ENTERPRISE COBOL INTERFACE--          *
*****
*
*          FLMTRNSL  CALLNAM='ENTERPRISE COBOL COMPILER',          C
*                   FUNCTN=BUILD,                        C
*                   COMPILE=IGYCRCTL,                    C
*                   DSNAME=IGY.SIGYCOMP,                 C
*                   VERSION=3.1,                        C
*                   GOODRC=0,                            C
*                   PORDER=1,                            C
*                   OPTIONS=(XREF,LIB,APOST,NODYNAM,LIST,NONUMBER,NOSEQ)
*
*****
*          --DDNAME ALLOCATION--          *
*****
*
*          FLMALLOC  IOTYPE=O,DDNAME=SYSLIN,KEYREF=OBJ,          C
*                   RECNUM=5000,DFLTYP=OBJ
*
*          FLMALLOC  IOTYPE=I,DDNAME=SYSLIB,KEYREF=SINC
*
*          FLMALLOC  IOTYPE=S,DDNAME=SYSIN,KEYREF=SINC,RECNUM=2000
*
*          FLMALLOC  IOTYPE=W,DDNAME=SYSUT1,RECNUM=5000
*
*          FLMALLOC  IOTYPE=W,DDNAME=SYSUT2,RECNUM=5000
*
*          FLMALLOC  IOTYPE=W,DDNAME=SYSUT3,RECNUM=5000
*
*          FLMALLOC  IOTYPE=W,DDNAME=SYSUT4,RECNUM=5000
*
*          FLMALLOC  IOTYPE=W,DDNAME=SYSUT5,RECNUM=5000
*
*          FLMALLOC  IOTYPE=W,DDNAME=SYSUT6,RECNUM=5000
*
*          FLMALLOC  IOTYPE=W,DDNAME=SYSUT7,RECNUM=5000
*
*          FLMALLOC  IOTYPE=A,DDNAME=SYSTEM
*                   FLMCPYLB  NULLFILE
*
*          FLMALLOC  IOTYPE=A,DDNAME=SYSPUNCH
*                   FLMCPYLB  NULLFILE
*
*          FLMALLOC  IOTYPE=O,DDNAME=SYSPRINT,KEYREF=LIST,          C
*                   RECFM=FBA,LRECL=133,                C
*                   RECNUM=50000,PRINT=Y,DFLTYP=LIST

```

- Change the caption for the above figure to "Figure 17. Enterprise COBOL Language Definition Example".
- Replace the 4 bullet points following Figure 17 from:
 - The name of the compiler: COBOL.

- The name of the compiler load module: IKFCBL00.
- The version of the compiler: 1.0.
- The compiler options: DMA, PRI, SIZE=512K, APOS, CNT=77, BUF=30K, OPT, XREF

to:

- The name of the compiler: ENTERPRISE COBOL.
- The name of the compiler load module: IGYCRCTL.
- The version of the compiler: 3.3.1.
- The compiler options:
XREF,LIB,APOST,NODYNAM,LIST,NONUMBER,NOSEQ.

"Step 9: Assemble and link the project definition" section

"Assemble and link example" subsection

Replace the example shown with the following:

```
//jobname JOB (wkpkg,dpt,bin),'name'
//* code additional JOBCARD statements here
//*
//ASMPROJ PROC PROJID=,PROJDEF=
/*-----*
/* ASSEMBLE AND LINK A PROJECT DEFINITION *
/* *
/* PROC PARAMETERS: *
/* *
/* PROJID - HIGH-LEVEL QUALIFIER FOR PROJECT *
/* PROJDEF - PROJECT DEFINITION MEMBER NAME *
/* *
/* NOTE: MODIFY SYSLIB DSNAMES TO GET THE SCLM RELEASE MACROS *
/* AND ANY LANGUAGE DEFINITIONS YOU NEED. *
/*-----*
//ASM EXEC PGM=ASMA90,REGION=4000K,PARM=OBJECT
//SYSLIB DD DSN=&PROJID..PROJDEFS.SOURCE,DISP=SHR
// DD DSN=ISP.SISPMACS,DISP=SHR
//SYSPRINT DD SYSOUT=H
//SYSPUNCH DD DUMMY
//SYSIN DD DSN=&PROJID..PROJDEFS.SOURCE(&PROJDEF),DISP=SHR
//SYSUT1 DD UNIT=SYSALLDA,SPACE=(CYL,(2,2))
//SYSLIN DD DSN=&&INT,DISP=(,PASS),UNIT=SYSDA,SPACE=(CYL,(5,5,0)),
// DCB=(BLKSIZE=400)
/*-----*
//LINK EXEC PGM=IEWL,PARM='RENT,LIST,MAP',REGION=512K
//SYSPRINT DD SYSOUT=H
//SYSLIN DD DSN=&&INT,DISP=(OLD,DELETE)
//SYSUT1 DD UNIT=SYSALLDA,SPACE=(CYL,(2,2)),DISP=NEW
//SYSLMOD DD DISP=SHR,DSN=&PROJID..PROJDEFS.LOAD(&PROJDEF)
// PEND
/*-----*
//ASMLINK EXEC PROC=ASMPROJ,PROJID=SCLM,PROJDEF=SCLM
```

"Project manager scenario" section

"Prerequisites for defining an SCLM project" subsection

Replace:

- PL/I Optimizing Compiler IEL0AA Version 4.0 or equivalent. (Optional if defining the project with the SCLM sample project utility.)

with:

- Enterprise PL/I for z/OS IBMZPLI Version 3.3.0 or equivalent.
- Enterprise COBOL for z/OS IGYCRCTL Version 3.3.1 or equivalent.
- Fault Analyzer. Version 6.1 or equivalent.

"Example project overview" subsection

Replace the following paragraph (and Note):

"Figure 19 on page 43 shows six modules in the hierarchy: FLM01MD1, FLM01MD2, FLM01MD3, FLM01MD4, FLM01MD5, and FLM01MD6. These are the programs that the developers edit in order to install fixes and new features.

- FLM01MD2 is written in PL/I and uses the PL/I optimization compiler.

Note: Module FLM01MD2 and the language definition for the PLI Optimizing Compiler are not included if the project is defined using the SCLM sample project utility."

with:

"Figure 19 on page 43 shows nine modules in the hierarchy: FLM01AD9, FLM01CD7, FLM01CD8, FLM01MD1, FLM01MD2, FLM01MD3, FLM01MD4, FLM01MD5, and FLM01MD6. These are the programs that the developers edit in order to install fixes and new features.

- FLM01MD2 is written in PL/I and uses the Enterprise PL/I compiler. FLM01MD2 includes the FLM01IIN copybook.

Note: Module FLM01MD2, copybook FLM01IIN and the language definition for the Enterprise PL/I Compiler are not included if the project is defined using the SCLM sample project utility and the first optional compiler field was not selected.

- Module FLM01AD9 is written in High Level Assembler. FLM01AD9 includes the FLM01INC copybook.

Note: Module FLM01AD9 and copybook FLM01INC will be included if the project is defined using the SCLM sample project utility but no side file will be generated if the Fault Analyzer field is not selected.

- FLM01CD7 and FLM01CD8 are written in COBOL and uses the Enterprise COBOL compiler. FLM01CD7 includes FLM01CIN copybook. FLM01CD8 includes the FLM01CDT copybook.

Note: Modules FLM01CD7, FLM01CD8, copybooks FLM01CIN, FLM01CDT and the language definition for the Enterprise COBOL Compiler are not included if the project is defined using the SCLM sample project utility and the second optional compiler field was not selected."

Replace the following two bullet points (and their respective Notes):

- FLM01AP1 does not directly call any language translators. It references other architecture members. The Build process creates the load modules FLM01LD1, FLM01LD2, FLM01LD3, and FLM01LD4.

Note: Load module FLM01LD2 is not created if the project is defined using the SCLM sample project utility.

- FLM01AP1, FLM01SB1, and FLM01SB2 are high-level architecture members. They do not call any language translators. FLM01LD1, FLM01LD2, FLM01LD3, and FLM01LD4 are LEC architecture members. FLM01CMD is a CC architecture member, and FLM01ARH is an architecture member that is directly copied into FLM01LD3 and FLM01LD4.

Note: Architecture member FLM01LD2 is not included if the project is defined using the SCLM sample project utility.

With:

- "FLM01AP1 does not directly call any language translators. It references other architecture members. The Build process creates the load modules FLM01LD7, FLM01LD9, FLM01LD1, FLM01LD2, FLM01LD3, and FLM01LD4.

Notes:

1. Load module FLM01LD2 is not created if the project is defined using the SCLM sample project utility and the first optional compiler field was not selected.
 2. Load module FLM01LD7 is not created if the project is defined using the SCLM sample project utility and the second optional compiler field was not selected.
- FLM01AP1, FLM01SB1, and FLM01SB2 are high-level architecture members. They do not call any language translators. FLM01LD7, FLM01LD9, FLM01LD1, FLM01LD2, FLM01LD3, and FLM01LD4 are LEC architecture members. FLM01CMD and FLM01CM9 are CC architecture members, and FLM01ARH is an architecture member that is directly copied into FLM01LD3 and FLM01LD4.

Notes:

1. Architecture member FLM01LD2 is not included if the project is defined using the SCLM sample project utility and the first optional compiler field was not selected.
2. Architecture member FLM01LD7 is not included if the project is defined using the SCLM sample project utility and the second optional compiler field was not selected."

Replace Figure 19 (Example Project Architecture) and the Note that follows with the following:

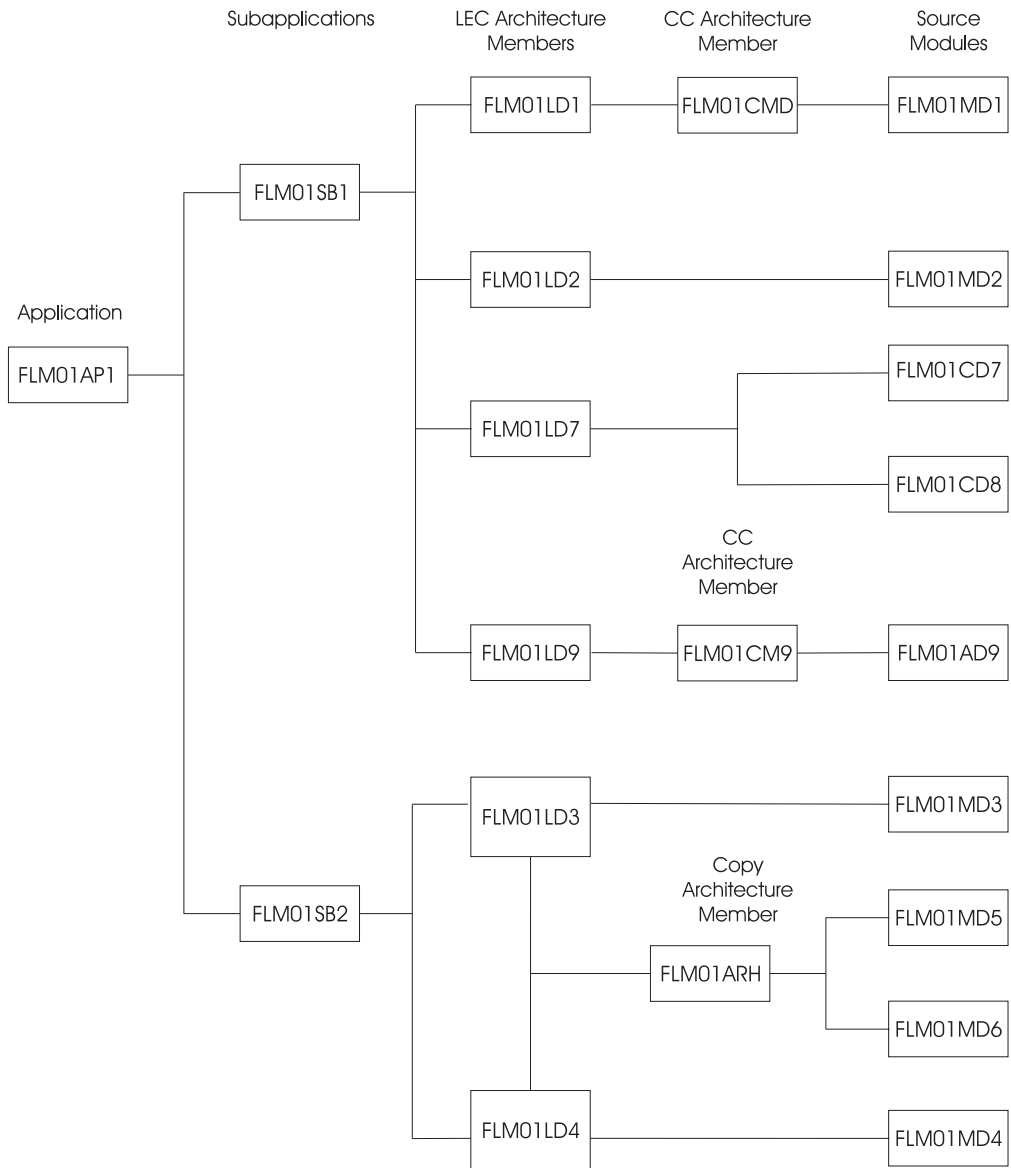


Figure 9. Example Project Architecture

Notes:

1. Source module FLM01MD2 and architecture member and load module FLM01LD2 are not included if the project was defined using the SCLM sample project utility (Option 10.7) and the first optional compiler field was not selected.
2. Source module FLM01CD7, FLM01CD8 and architecture member and load module FLM01LD7 are not included if the project was defined using the SCLM sample project utility (Option 10.7) and the second optional compiler field was not selected.

"Preparing the example project hierarchy" subsection

Step 6 Replace the existing text with:

Use the ISPF Move/Copy Utility to copy the following members from ISP.SISPSAMP into PROJ1.PROJDEFS.SOURCE: FLM01ASF,

FLM01ASM, FLM01CBE, FLM01PLE, FLM01PRJ, FLM01SCR,
FLM01370, FLM02ALL, and FLM02ACT.

Notes:

1. If you are not using Fault Analyzer, do not copy FLM01ASF.
2. If you are not using Fault Analyzer, delete the following lines from FLM01PRJ:
 - IDILANGX FLMTYPE
 - COPY FLM01ASF

Step 7 Add the following note:

Note: If you are not using Fault Analyzer, remove the following step from FLM02ALL:

```
//*  
//STEP3ID EXEC PGM=IEFBRI4  
//DEV1 DD DSN=project.DEV1.IDILANGX,DISP=(NEW,CATLG),UNIT=SYSDA,  
// DCB=(RECFM=VB,LRECL=1562,BLKSIZE=0),SPACE=(27998,(5,20,10))  
//DEV2 DD DSN=project.DEV2.IDILANGX,DISP=(NEW,CATLG),UNIT=SYSDA,  
// DCB=(RECFM=VB,LRECL=1562,BLKSIZE=0),SPACE=(27998,(5,20,10))  
//TEST DD DSN=project.TEST.IDILANGX,DISP=(NEW,CATLG),UNIT=SYSDA,  
// DCB=(RECFM=VB,LRECL=1562,BLKSIZE=0),SPACE=(27998,(5,20,10))  
//RELEASE DD DSN=project.RELEASE.IDILANGX,DISP=(NEW,CATLG),UNIT=SYSDA,  
// DCB=(RECFM=VB,LRECL=1562,BLKSIZE=0),SPACE=(27998,(5,20,10))
```

Step 8.c

Replace:

```
PROJ1.DEV1.SOURCLST  
PROJ1.DEV2.SOURCLST  
PROJ1.TEST.SOURCLST  
PROJ1.RELEASE.SOURCLST
```

with:

```
PROJ1.DEV1.LIST  
PROJ1.DEV2.LIST  
PROJ1.TEST.LIST  
PROJ1.RELEASE.LIST
```

Step 8.f

Replace:

"...format FB, LRECL 121:"

with:

"...format FBA, LRECL 121:"

After Step8.f, add an extra step:

g. Using the ISPF Data Set Utility, allocate the following partitioned data set with space in blocks (300,100), with 10 directory blocks, and with record format VB, LRECL 1562:

```
PROJ1.DEV1.IDILANGX  
PROJ1.DEV2.IDILANGX  
PROJ1.TEST.IDILANGX  
PROJ1.RELEASE.IDILANGX
```

This partitioned data set contains side files generated by Fault Analyzer.

Note: If you are not using Fault Analyzer, do not allocate PROJ1.*.IDILANGX datasets.

Step 11

Replace the first two paragraphs with:

Using ISPF Edit, edit PROJ1.PROJDEFS.SOURCE(FLM01ASM). Change all system macro library references to the library of macros at your location.

You must change the members FLM01ASF, FLM01CBE, FLM01PLE, FLM01SCR, and FLM01370 so that libraries, assemblers, and assembler options match the libraries and products in use at your location. The changes are specified in the samples delivered.

Step 15

Replace the existing text with the following:

If you used the sample JCL in the "Assemble and link example" on page 41, make sure that member PROJ1.PROJDEFS.LOAD(PROJ1) exists.

Otherwise use the ISPF Foreground Linkage Editor to link-edit PROJ1.PROJDEFS.OBJ(PROJ1). This constructs the load module PROJ1.PROJDEFS.LOAD(PROJ1) that is executed by SCLM to control the library.

Verify that the link occurred without errors.

Step 16

Replace:

FLM01EQU, FLM01MD1, FLM01MD2, FLM01MD3, FLM01MD4, FLM01MD5, and FLM01MD6.

with:

FLM01AD9, FLM01CDT, FLM01CD7, FLM01CD8, FLM01EQU, FLM01CIN, FLM01IIN, FLM01INC, FLM01MD1, FLM01MD2, FLM01MD3, FLM01MD4, FLM01MD5, and FLM01MD6.

Step 17

Replace:

FLM01AP1, FLM01ARH, FLM01CMD, FLM01LD1, FLM01LD2, FLM01LD3, FLM01LD4, FLM01SB1, and FLM01SB2.

with:

FLM01AP1, FLM01ARH, FLM01CMD, FLM01CM9, FLM01LD1, FLM01LD2, FLM01LD3, FLM01LD4, FLM01LD7, FLM01LD9, FLM01SB1, and FLM01SB2.

Note: If you are not using Fault Analyzer, edit FLM01CM9 as follows:

- Delete line OUT2 FLM01AD9 IDILANGX
- Change PARM1 to PARM

"Understanding the sample project definition" subsection

- In step 1, for FLMTYPE in the table of macros, replace:

SOURCLST listings from compilers and assemblers

with:

LIST listings from compilers and assemblers

IDILANGX Fault Analyzer datasets

Note: If you are not using Fault Analyzer, make sure that you have edited FLM01PRJ as specified in step 6 of "Preparing the example project hierarchy".

- In step 1, replace the last line:
"...to be used (for example, architecture definition, ASM, PL/I, and link-edit)."
with:
"...to be used (for example, architecture definition, ASM, COBOL, PL/I, and link-edit)."
- In step 2, replace the list of PROJ1.PROJDEFS.SOURCE members with:

FLM01ASF	ASMF language definition
FLM01ASM	ASM language definition
FLM01CBE	COBE language definition
FLM01PLE	PLIE language definition
FLM01370	linkage editor language definition

Note: FLM01ASF will be displayed only if you are using Fault Analyzer.

"Preparing the example project data" subsection

Step3 Replace:

3. From the Utilities panel, select the Migration option. Type:

SOURCE	in the Type field
FLM01MD2 (the PL/I module)	in the Member field
1	in the Mode field
PLI0	in the Language field
1	in the Process field
1	in the Messages field
4	in the Listings field

with:

- 3.1 Skip this step if you are not using Fault Analyzer. Otherwise from the Utilities panel, select the Migration option. Type:

SOURCE	in the Type field
FLM01AD9 (the ASM module)	in the Member field
1	in the Mode field
HLAF	in the Language field
1	in the Process field
1	in the Messages field
4	in the Report field
4	in the Listings field

Press Enter to begin processing. The migration utility registers new modules (in this case, FLM01AD9) into an SCLM library by creating accounting records for them.

- 3.2 Skip this step if you are not using Fault Analyzer. Otherwise from the Utilities panel, select the Migration option. Type:

SOURCE	in the Type field
FLM01INC (the ASM copybook)	in the Member field
1	in the Mode field
HIAF	in the Language field
1	in the Process field
1	in the Messages field
4	in the Report field
4	in the Listings field

Press Enter to begin processing. The migration utility registers new modules (in this case, FLM01INC) into an SCLM library by creating accounting records for them.

- 3.3 From the Utilities panel, select the Migration option. Type:

SOURCE	in the Type field
FLM01C* (the COBOL copybook)	in the Member field
1	in the Mode field
COBE	in the Language field
1	in the Process field
1	in the Messages field
4	in the Report field
4	in the Listings field

Press Enter to begin processing. The migration utility registers new modules (in this case, FLM01CIN, FLM01CDT, FLM01CD7, FLM01CD8) into an SCLM library by creating accounting records for them.

- 3.4 From the Utilities panel, select the Migration option. Type:

SOURCE	in the Type field
FLM01IIN (the PL/I copybook)	in the Member field
1	in the Mode field
PLIE	in the Language field
1	in the Process field
1	in the Messages field
4	in the Report field
4	in the Listings field

Press Enter to begin processing. The migration utility registers new modules (in this case, FLM01IIN) into an SCLM library by creating accounting records for them.

- 3.5 From the Utilities panel, select the Migration option. Type:

SOURCE	in the Type field
FLM01MD2 (the PL/I copybook)	in the Member field
1	in the Mode field
PLIE	in the Language field
1	in the Process field
1	in the Messages field
4	in the Report field
4	in the Listings field

Press Enter to begin processing. The migration utility registers new modules (in this case, FLM01MD2) into an SCLM library by creating accounting records for them.

Step4 Replace:

Because FLM01MD2 was migrated first, SCLM does not migrate it again when an * is specified.

with:

Because FLM01CDT, FLM01CIN, FLM01IIN, FLM01CD7, FLM01CD8, FLM01MD2, FLM01INC, and FLM01AD9 were migrated first, SCLM does not migrate them again when an * is specified.

Note: FLM01INC and FLM01AD9 would be migrated already only if you are using Fault Analyzer.

Chapter 3, "Additional project manager tasks"

"Implementing package backout" section

Figure 23 (Sample project definition)

Replace:

```
SOURCLST FLMTYPE
```

with:

```
LIST FLMTYPE
```

Chapter 5, "Language definition considerations"

"Converting JCL to SCLM language definitions" section

"Converting JCL cards to SCLM macro statements" subsection,:

"Sample JCL conversion" subsection:

- Replace steps 2 and 3 with:
 2. Job step TRN executes a program called DFHECP1\$, the CICS preprocessor for COBOL.
SCLM uses the COMPILE=DFHECP1\$ statement on the FLMTRNSL macro.
 3. The STEPLIB line in job step TRN tells the job where to find the program DFHECP1\$.
SCLM uses the DSNNAME option on the FLMTRNSL macro. Both the STEPLIB and DSNNAME point to the same data set, CICS.TS31.CICS.SDFHLOAD.
- Replace step 8 with:
 8. The job step COB executes (EXEC PGM=) a program called IGYCRCTL, the compiler for COBOL.
SCLM uses the COMPILE=IGYCRCTL statement on the FLMTRNSL macro.
- Replace steps 10 and 11 with:
 10. This job has conditional execution for the COB step via the COND(5,LE) JCL command. The COB step will not execute if the return code of the TRN step is greater than 4.

SCLM sets the GOODRC keyword parameter for the TRN step (CICS preprocessor) equal to 4. Build halts execution of all translators following the TRN step in the language definition if the return code from the TRN step is greater than 4.

11. The STEPLIB statement in job step COB tells the job where to find the program IGYCRCTL.

SCLM uses the DSNNAME= option on the FLMTRNSL macro. Both the STEPLIB and DSNNAME point to the same data set, IGY.SIGYCOMP.

- Replace step 13 with:

13. For each system library specified for the SYSLIB DD, the language definition has an FLMSYSLB macro. In this case both CICS.TS31.CICS.SDFHCOB and CICS.TS31.CICS.SDFHMAC are specified.

- Steps 16–20: replace all occurrences of "OS COBOL" with "COBOL".

- Replace steps 21 and 22 with:

21. The COB step allocates SYSUT5 as a temporary work file for the COBOL compiler.

SCLM's language definition uses an FLMALLOC macro with IOTYPE=W to perform the same task. This must be the 13th file provided to the COBOL compiler because we are using a ddname substitution list.

22. The COB step allocates SYSUT6 as a temporary work file for the COBOL compiler.

SCLM's language definition uses an FLMALLOC macro with IOTYPE=W to perform the same task. This must be the 14th file provided to the COBOL compiler because we are using a ddname substitution list.

23. The COB step allocates SYSUT7 as a temporary work file for the COBOL compiler.

SCLM's language definition uses an FLMALLOC macro with IOTYPE=W to perform the same task. This must be the 15th file provided to the COBOL compiler because we are using a ddname substitution list.

24. SCLM language definition only The language definition uses PORDER=3 for the COBOL compiler step (COB) to use a ddname substitution list. A ddname substitution list provides an ordered list (defined by the translator) of ddnames such that the position of a ddname in the list, and not the actual ddname, is used by the translator for a specific file.

The input file for the compiler must be the output file from the CICS preprocessor. The ddname assigned in the TRN step is SYSPUNCH. Because this file has already been allocated to SYSPUNCH, another way (besides ddname) is needed to pass this file as the input to the compiler. By using PORDER=3, SCLM passes all the files that can be used by the COBOL compiler in the order specified for this compiler. To use PORDER=3, a specific parameter string must be built. The language definition must have an FLMALLOC macro for each of these parameters.

Those FLMALLOCs that are tagged for STEP 24 are not applicable for the COBOL compiler. SCLM places 8 bytes of hexadecimal zeros into the ddname substitution list for each FLMALLOC with IOTYPE=N.

- Replace figure 45 (JCL: Invoke COBOL Preprocessor and Compiler) with:

```

//USERIDC JOB (AS05CR,T12,C531),'USERID',NOTIFY=&SYSUID,CLASS=A,
//          MSGCLASS=0,MSGLEVEL=(1,1)
//*
//*   THIS PROCEDURE CONTAINS 2 STEPS
//*   1. EXEC THE CICS PREPROCESSOR
//*   2. EXEC THE Enterprise COBOL COMPILER
//*
//*   CHANGE THE JOB NAME AND THE ACCOUNTING INFORMATION TO MEET THE
//*   REQUIREMENTS OF YOUR INSTALLATION.
//*
//*   CHANGE 'PROGNAME' TO THE NAME OF THE CICS/COBOL PROGRAM YOU
//*   WANT TO COMPILE. CHANGE 'USERID' TO YOUR USERID.
//*
//*   CHANGE 'DEVL' TO THE GROUP THAT CONTAINS THE PROGRAM TO BE
//*   COMPILED
//*
//* STEP 1: TRN STATEMENT; STEP 2: EXEC PGM STATEMENT
//*
//TRN EXEC PGM=DFHECP1$,
//          REGION=2048K
//*
//* STEP 3: STEPLIB STATEMENT
//*
//STEPLIB DD DSN=CICS.TS31.CICS.SDFHLOAD,DISP=SHR
//*
//* STEP 4: SYSIN STATEMENT
//*
//SYSIN DD DSN=USERID.DEVL.SOURCE(PROGNAME),DISP=SHR
//*
//* STEP 5: SYSPRINT STATEMENT
//*
//SYSPRINT DD SYSOUT=A
//*
//* STEP 6: SYSPUNCH STATEMENT
//*
//SYSPUNCH DD DSN=&&SYSCIN,
//          DISP=(,PASS),UNIT=SYSDA,
//          DCB=BLKSIZE=400,
//          SPACE=(400,(400,100))
//*
//* STEP 7: COB STATEMENT; STEP 8: EXEC PGM STATEMENT

```

Figure 10. JCL: Invoke COBOL Preprocessor and Compiler (Part 1 of 2)

```

/**
//COB EXEC PGM=IGYCRCTL,REGION=2048K,COND=(5,LE),
//          PARM=' RENT,NODYNAM,LIB'
/**
/** STEP 11: STEPLIB STATEMENT
/**
//STEPLIB DD DSN=IGY.SIGYCOMP,DISP=SHR
/**
/** STEP 12: SYSLIB STATEMENT; STEP 13: DD STATEMENT
/**
//SYSLIB DD DSN=CICS.TS31.CICS.SDFHCOB,DISP=SHR
//          DD DSN=CICS.TS31.CICS.SDFHMAC,DISP=SHR
/**
/** STEP 14: SYSPRINT STATEMENT
/**
//SYSPRINT DD SYSOUT=0
/**
/** STEP 15: SYSIN STATEMENT
/**
//SYSIN DD DSN=&&SYSCIN,DISP=(OLD,DELETE)
/**
/** STEP 16: SYSLIN STATEMENT
/**
//SYSLIN DD DSN=USERID.DEVLEV.OBJ(PROGNAME),DISP=SHR
/**
/** STEP 17: SYSUT1 STATEMENT
/**
//SYSUT1 DD UNIT=SYSDA,SPACE=(460,(350,100))
/**
/** STEP 18: SYSUT2 STATEMENT
/**
//SYSUT2 DD UNIT=SYSDA,SPACE=(460,(350,100))
/**
/** STEP 19: SYSUT3 STATEMENT
/**
//SYSUT3 DD UNIT=SYSDA,SPACE=(460,(350,100))
/**
/** STEP 20: SYSUT4 STATEMENT
/**
//SYSUT4 DD UNIT=SYSDA,SPACE=(460,(350,100))
/**
/** STEP 21: SYSUT5 STATEMENT
/**
//SYSUT5 DD UNIT=SYSDA,SPACE=(460,(350,100))
/**
/** STEP 22: SYSUT5 STATEMENT
/**
//SYSUT6 DD UNIT=SYSDA,SPACE=(460,(350,100))
/**
/** STEP 23: SYSUT5 STATEMENT
/**
//SYSUT7 DD UNIT=SYSDA,SPACE=(460,(350,100))

```

Figure 10. JCL: Invoke COBOL Preprocessor and Compiler (Part 2 of 2)

- Replace figure 46 (SCLM Language Definition: Invoke COBOL Preprocessor and Compiler) with:

```

*****
* SCLM LANGUAGE DEFINITION FOR
* COBOL WITH CICS 3.1 PREPROCESSOR
*
* CICS OUTPUT IS PASSED VIA THE CICSTRAN DD ALLOCATION TO COBOL.
*
* POINT THE FLMSYSLB MACRO(S) AT ALL 'STATIC' COPY DATASETS.
* CUSTOMIZE THE 'OPTIONS' AND 'GOODRC' FIELDS TO YOUR STANDARDS.
* ADD THE 'DSNAME' FIELD IF THE TRANSLATOR IS IN A PRIVATE LIBRARY.
* WHEN A NEW TRANSLATOR VERSION REQUIRES TOTAL RECOMPILATION FOR THIS
* LANGUAGE, THE 'VERSION' FIELD ON FLMLANGL SHOULD BE CHANGED.
*****
*
*STEP 13
COBCICS  FLMSYSLB CICS.TS31.CICS.SDFHCOB
          FLMSYSLB CICS.TS31.CICS.SDFHMAC
*
          FLMLANGL LANG=COBCICS,VERSION=CICS31,ALCSYSLB=Y
*
* PARSER TRANSLATOR
*
          FLMTRNSL CALLNAM='SCLM COBOL PARSE',
          FUNCTN=PARSE,
          COMPILE=FLMLPCBL,
          PORDER=1,
          OPTIONS=(@@FLMLIS,@@FLMSTP,@@FLMSIZ,)
*      (* SOURCE *)
          FLMALLOC IOTYPE=A,DDNAME=SOURCE
          FLMCPYLB @@FLMDSN(@@FLMMBR)
*
* BUILD TRANSLATORS
*   - CICS PRECOMPILE - STEP NAME TRN
*
* STEP 1
          FLMTRNSL CALLNAM='CICS PRE-COMPILE',
          FUNCTN=BUILD,
* STEP 2
          COMPILE=DFHECP1$,
* STEP 3 (* STEPLIB *)
          DSNAME=CICS.TS31.CICS.SDFHLOAD,
          VERSION=3.1,
* STEP 10 (* COND *)
          GOODRC=4,
          PORDER=1,
          OPTIONS=(NOSEQ)
* STEP 4 (* SYSIN *)
          FLMALLOC IOTYPE=S,KEYREF=SINC,RECFM=FB,LRECL=80,
          DDNAME=SYSIN
* STEP 5 (* SYSPRINT *)
          FLMALLOC IOTYPE=0,RECFM=FBA,LRECL=121,
          RECNUM=35000,PRINT=Y,DDNAME=SYSPRINT
*
* STEP 6 (* SYSPUNCH *)
          FLMALLOC IOTYPE=W,RECFM=FB,LRECL=80,
          RECNUM=5000,DDNAME=SYSPUNCH
*
* STEP 7 (*COBOL INTERFACE - STEP NAME COB *)
* STEP 8
          FLMTRNSL CALLNAM='COBOL COMPILE',
          FUNCTN=BUILD,
          COMPILE=IGYCRCTL,
* STEP 11 (* STEPLIB *)
          DSNAME=IGY.SIGYCOMP,
          VERSION=3.3.1,
          GOODRC=4,

```

Figure 11. SCLM Language Definition: Invoke COBOL Preprocessor and Compiler (Part 1 of 2)

```

* STEP 24
      PORDER=3,
* STEP 9 (* PARS * )
      OPTIONS=( RENT,NODYNAM,LIB)
* DDNAME ALLOCATIONS
* STEP 16
* 1 (* SYSLIN *)
      FLMALLOC IOTYPE=0,KEYREF=OBJ,RECFM=FB,LRECL=80,
      RECNUM=5000,DFLTYP=OBJ
* STEP 24
* 2 (* N/A *)
      FLMALLOC IOTYPE=N
* STEP 24
* 3 (* N/A *)
      FLMALLOC IOTYPE=N
* STEP 12; STEP 13
* 4 (* SYSLIB *)
      FLMALLOC IOTYPE=I,KEYREF=SINC
* STEP 15
* 5 (* SYSIN *)
      FLMALLOC IOTYPE=U,KEYREF=SINC,DDNAME=SYSPUNCH
* STEP 14
* 6 (* SYSPRINT *)
      FLMALLOC IOTYPE=0,KEYREF=LIST,RECFM=FBA,LRECL=133,
      RECNUM=25000,PRINT=Y,DFLTYP=LIST
* STEP 24
* 7 (* SYSPUNCH *)
      FLMALLOC IOTYPE=N
      FLMCPYLB NULLFILE
* STEP 17
* 8 (* SYSUT1 *)
      FLMALLOC IOTYPE=W,RECFM=FB,LRECL=80,RECNUM=5000
* STEP 18
* 9 (* SYSUT2 *)
      FLMALLOC IOTYPE=W,RECFM=FB,LRECL=80,RECNUM=5000
* STEP 19
* 10 (* SYSUT3 *)
      FLMALLOC IOTYPE=W,RECFM=FB,LRECL=80,RECNUM=5000
* STEP 20
* 11 (* SYSUT4 *)
      FLMALLOC IOTYPE=W,RECFM=FB,LRECL=80,RECNUM=5000
* STEP 24
* 12 (* SYSTEM *)
      FLMALLOC IOTYPE=N
      FLMCPYLB NULLFILE
* STEP 21
* 13 (* SYSUT5 *)
      FLMALLOC IOTYPE=W,RECFM=FB,LRECL=80,RECNUM=5000
* STEP 22
* 14 (* SYSUT6 *)
      FLMALLOC IOTYPE=W,RECFM=FB,LRECL=80,RECNUM=5000
* STEP 23
* 15 (* SYSUT7 *)
      FLMALLOC IOTYPE=W,RECFM=FB,LRECL=80,RECNUM=5000
* 5694-A01    COPYRIGHT IBM CORP 1980, 1989, 2007

```

Figure 11. SCLM Language Definition: Invoke COBOL Preprocessor and Compiler (Part 2 of 2)

Chapter 9, "Using SCLM functions"

"Utilities (option 3)" section

"Architecture Report Utility" subsection:

Figure 75, Architecture Report (FLMRA#P)

In Group field, replace:

USERID

with:

DEV1

Figure 76, Architecture report with cutoff of NONE (Part 2 of 3)

Replace:

CROSS REFERENCE FOR TYPE: SOURCLST

with:

CROSS REFERENCE FOR TYPE: LIST

"Export Utility" subsection:

Figure 78, SCLM Export Utility (FLMDXE#P)

In Group field, replace:

USERID

with:

DEV1

Figure 79, Export Report (Part 2 of 2)

Replace 3 occurrences of:

SOURCLST

with:

LIST

"Import Utility" subsection:

Figure 80, SCLM Import Utility (FLMDXE#P)

In Group field, replace:

USERID

with:

DEV1

Figure 81, Import Report (Part 2 of 2)

Replace 3 occurrences of:

SOURCLST

with:

LIST

"Unit of Work Utility" subsection:

Figure 101, UOW Member List panel

Replace with:

```

Menu  SCLM  Functions  Utilities  Help
-----
UOW Member List: SCLMTEST.DEV1.ARCHDEF - HIERARCHY VIEW -          Member 1 of 12

S=Sel/Edit A=Acct M=Map B=Browse D=Del E=Edit V=View C=Build P=Promote U=Upd
Z=Versions
Member  Status      Account  Language  Text      Chg Date  Chg Time
-      FLM01AP1          DEV1     ARCHDEF   DEV1      2006/12/11 12:54:25
-      FLM01ARH          DEV1     ARCHDEF   DEV1      2006/12/11 12:54:39
-      FLM01CMD          DEV1     ARCHDEF   DEV1      2006/12/11 12:54:52
-      FLM01CM9          DEV1     ARCHDEF   DEV1      2006/12/11 12:54:59
-      FLM01LD1          DEV1     ARCHDEF   DEV1      2006/12/11 12:55:19
-      FLM01LD2          DEV1     ARCHDEF   DEV1      2006/12/11 12:55:29
-      FLM01LD3          DEV1     ARCHDEF   DEV1      2006/12/11 12:55:35
-      FLM01LD4          DEV1     ARCHDEF   DEV1      2006/12/11 12:55:41
-      FLM01LD7          DEV1     ARCHDEF   DEV1      2006/12/11 12:55:49
-      FLM01LD9          DEV1     ARCHDEF   DEV1      2006/12/11 12:55:56
-      FLM01SB1          DEV1     ARCHDEF   DEV1      2006/12/11 12:56:03
-      FLM01SB2          DEV1     ARCHDEF   DEV1      2006/12/11 12:56:09
***** Bottom of data *****

Command ==>>> _____ Scroll ==>> PAGE
F1=Help      F3=Exit      F10=Actions  F12=Cancel
F10=Actions  F12=Cancel

```

Figure 12. UOW Member List panel

"Sample Project Utility (option 7)" section

In the third paragraph, replace:

You have a choice of including a PLI sample if you have the PLI Optimizing Compiler installed on your system.

with:

- If you have the Enterprise PL/I Compiler installed on your system, you have a choice of including a PL/I sample.
- If you have the Enterprise COBOL Compiler installed on your system, you have a choice of including a COBOL sample.
- If Fault Analyzer is installed on your system, you have a choice of generating a side file for member FLM01AD9.

Chapter 10, "Development scenario"

"Understanding the hierarchy and the SCLM main menu" section

- Replace the paragraph starting "The sample application is composed of..." and the Note that follows with:

"The sample application is composed of 9 programs that are used to build an application called FLM01AP1, as shown in Figure 112 on page 254. The programs are linked into 6 load modules. The 6 load modules are organized as 2 subapplications, which in turn are components of FLM01AP1.

Note: If neither the Enterprise PL/I Compiler nor Enterprise COBOL Compiler are included as a language in the sample project, the application consists of 6 programs linked into 4 load modules.

- Replace Figure Figure 112 (Application FLM01AP1) and the Note that follows with the following:

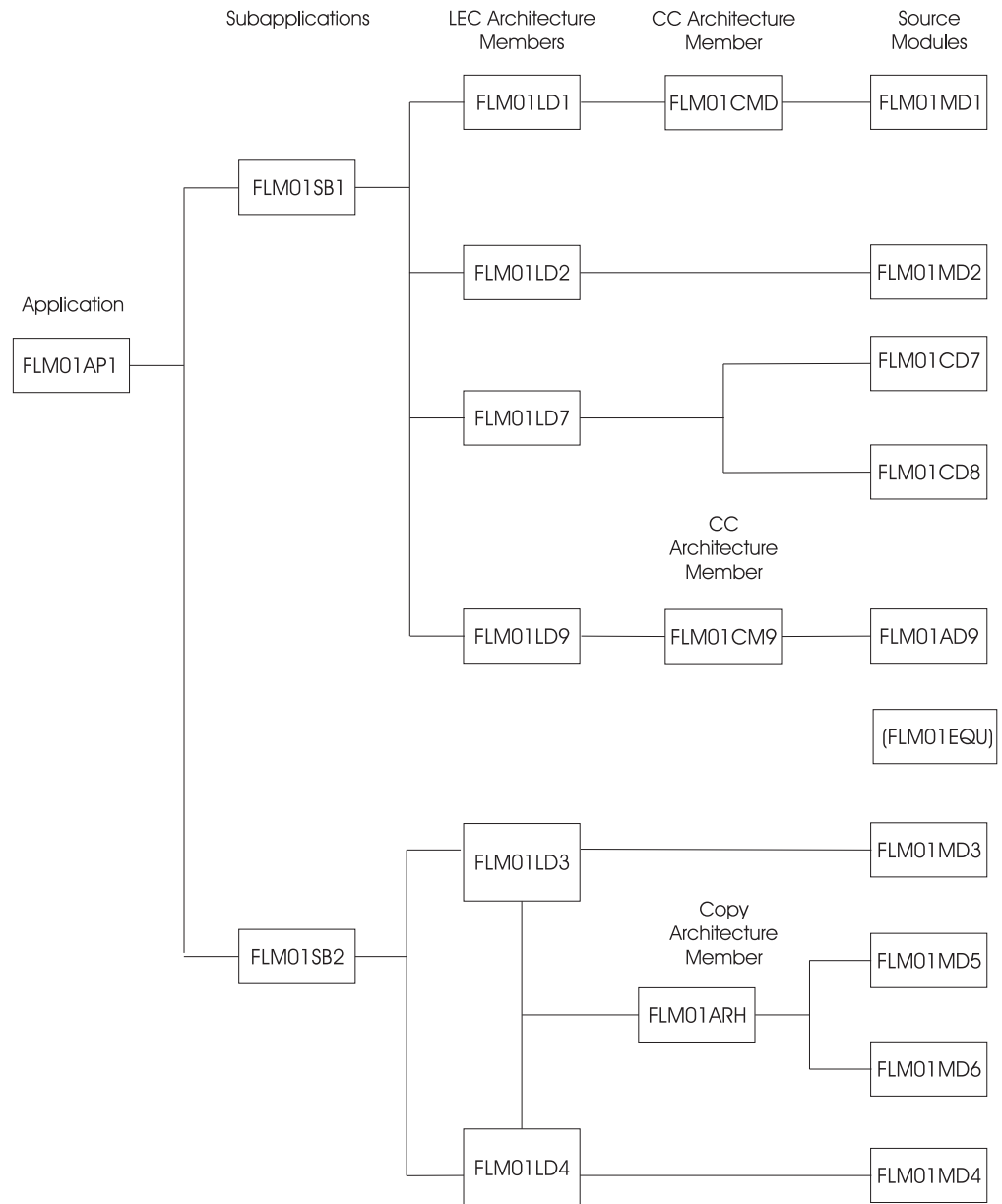


Figure 13. Application FLM01AP1

Notes:

1. Source module FLM01MD2 and architecture member FLM01LD2 are included only if Enterprise PL/I Compiler is included as a language if the sample is defined using the SCLM Sample Project utility (Option 10.7).
2. Source modules FLM01CD7, FLM01CD8, and architecture member FLM01LD7 are included only if Enterprise COBOL Compiler is included as a language if the sample is defined using the SCLM Sample Project utility (Option 10.7).

"Understanding the architecture definition" section

In the table of architecture members:

- Replace the table entry for FLM01SB1 HL with:

FLM01SB1	HL	References FLM01LD1, FLM01LD2, FLM01LD7, and FLM01LD9 with the INCL statement. A build performed on FLM01SB1 results in a complete build of the FLM01SB1 subapplication, if necessary. If the Enterprise PL/I Compiler is not included as a language in the sample project, FLM01SB1 does not reference FLM01LD2. If the Enterprise COBOL Compiler is not included as a language in the sample project, FLM01SB1 does not reference FLM01LD7.
----------	----	---

- Replace the table entry for FLM01LD2 LEC with:

FLM01LD2	LEC	Directs SCLM to build load module FLM01LD2 from the source FLM01MD2. The INCLD architecture statement is used to identify FLM01MD2 as the source. Note that LOAD, LMAP, and SOURCE are types identified by the FLMTYPE macro in the project definition. If the Enterprise PL/I Compiler is not included as a language in the sample project, FLM01LD2 is not included.
FLM01LD7	LEC	Directs SCLM to build load module FLM01LD7 from source FLM01CD7 and FLM01CD8. The INCLD architecture statement is used to identify FLM01CD7 and FLM01CD8 as the source. Note that LOAD, LMAP, and SOURCE are types identified by the FLMTYPE macro in the project definition. If the Enterprise COBOL Compiler is not included as a language in the sample project, FLM01LD7 is not included.
FLM01LD9	LEC	Directs SCLM to produce the load module and load map for FLM01LD9. The INCL statement references architecture member FLM01CM9. The PARM statements pass parameters to the SCLM BUILD translators.

- Replace the table entry for FLM01CMD CC with:

FLM01CMD	CC	Directs SCLM to produce object code from FLM01MD1. SINC identifies FLM01MD1 as the source member. Note that in addition to object code (OBJ), there is also source listing (LIST). OBJ and LIST are identified in the project definition with the FLMTYPE macro.
FLM01CM9	CC	Directs SCLM to produce object code from FLM01AD9. SINC identifies FLM01AD9 as the source member. Note that in addition to object code (OBJ), there is also source listing (LIST). OBJ and LIST are identified in the project definition with the FLMTYPE macro.

"Rebuilding the changed member" section

In step 2, after "Press Enter", replace the paragraph starting "Note the traversal..." with:

Note that:

- The traversal of the architecture. FLM01MD2 was not affected by the change to the FLM01EQU member and will not be recompiled. FLM01LD2, which contains only FLM01MD2, will not be relinked.
- FLM01CD7 was not affected by the change to the FLM01EQU member and will not be recompiled. FLM01LD7, which contains only FLM01CD7 and FLM01CD8, will not be relinked.

"Using the Database Contents Utility" section

- In step 5, replace the last paragraph (starting "Note that FLM01LD2...") with:
"Note that FLM01LD2, FLM01LD7, and FLM01LD8 do not appear in the DEV2 group. These modules were not affected by the changes to FLM01EQU."
- In step 6, replace the last paragraph (starting "Note that FLM01LD2...") with:
"Note that FLM01LD2, FLM01LD7, and FLM01LD8 does not appear in the DEV2 group. These modules were not affected by the changes to FLM01EQU."
- In step 7, replace the last paragraph (starting "Note that FLM01LD2...") with:
"Note that FLM01LD2, FLM01LD7, and FLM01LD8 does not appear in the DEV2 group. These modules were not affected by the changes to FLM01EQU."

Chapter 11, "Architecture definition"

- Replace the first paragraph (starting "The following application is composed of...") with:
"The following application is composed of two subapplications. The first subapplication consists of four load modules and the second subapplication consists of two load modules, that are composed of a series of object modules. Load module FLM01LD1, FLM01LD2, and FLM01LD9 contain one object module each, while FLM01LD3, FLM01LD4, and FLM01LD7 contain multiple object modules. Figure 114 on page 280 shows a diagram of the design of this application (FLM01AP1) and Figure 115 on page 281 shows the architecture members for the FLM01AP1 application."
- Replace figure 114 (Application FLM01AP1) with the following:

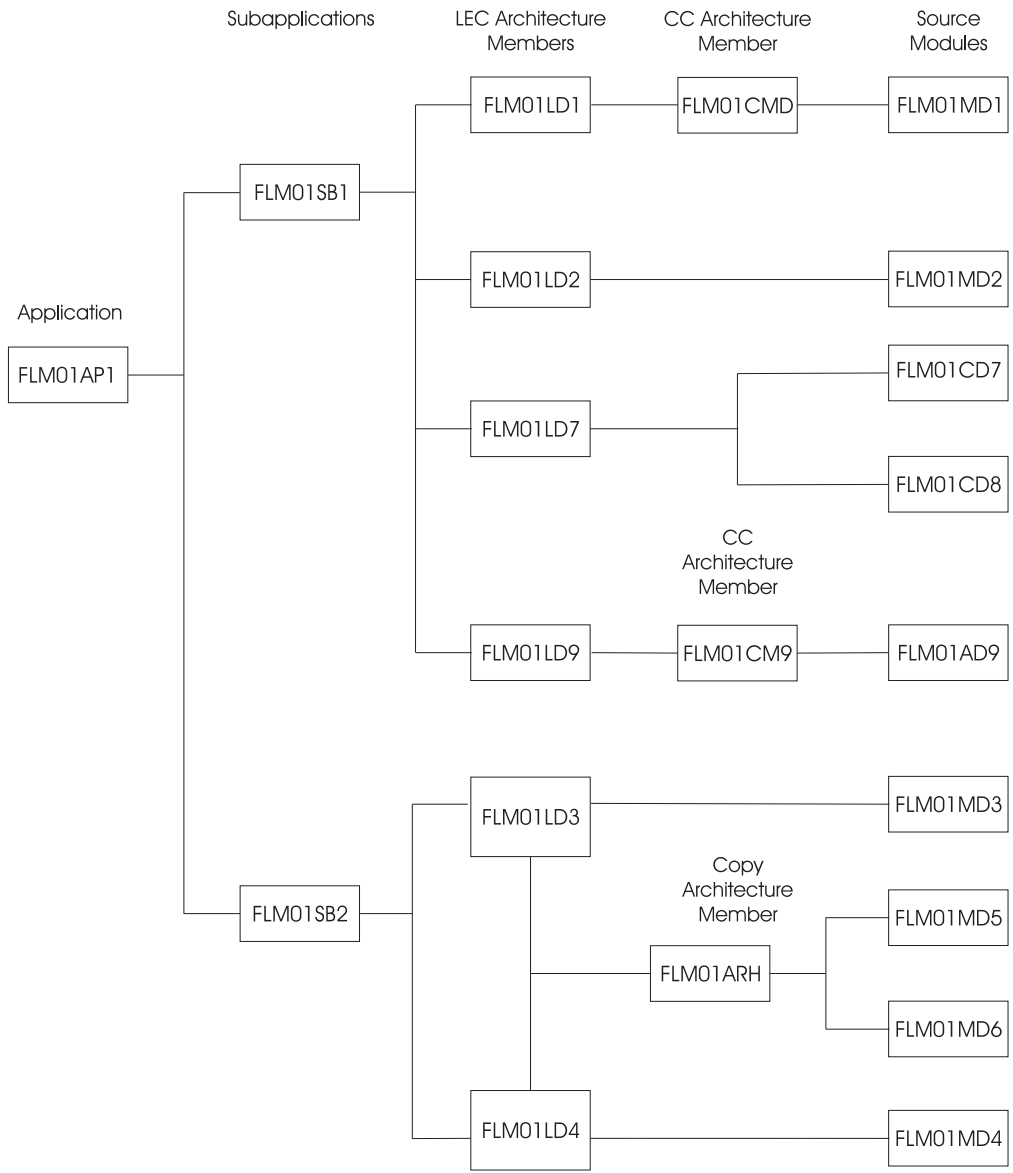
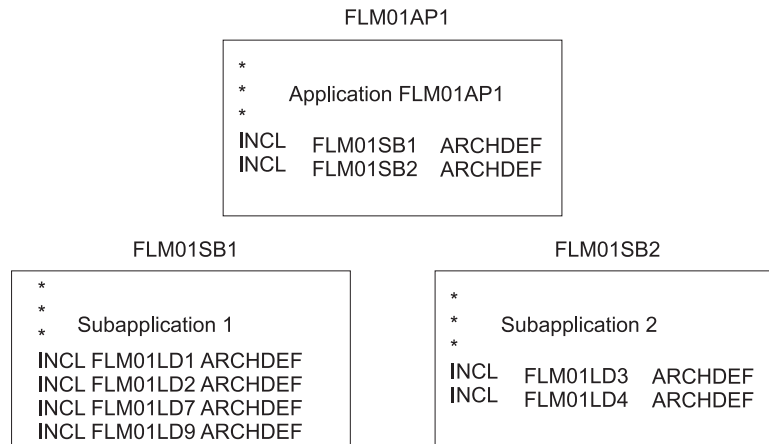


Figure 14. Application FLM01AP1

- Replace figure 115 (Architecture Members for Application Sample FLM01AP1 (Part 1 of 2) with the following:

High-Level Architecture Members



Linkedit Control Architecture Members

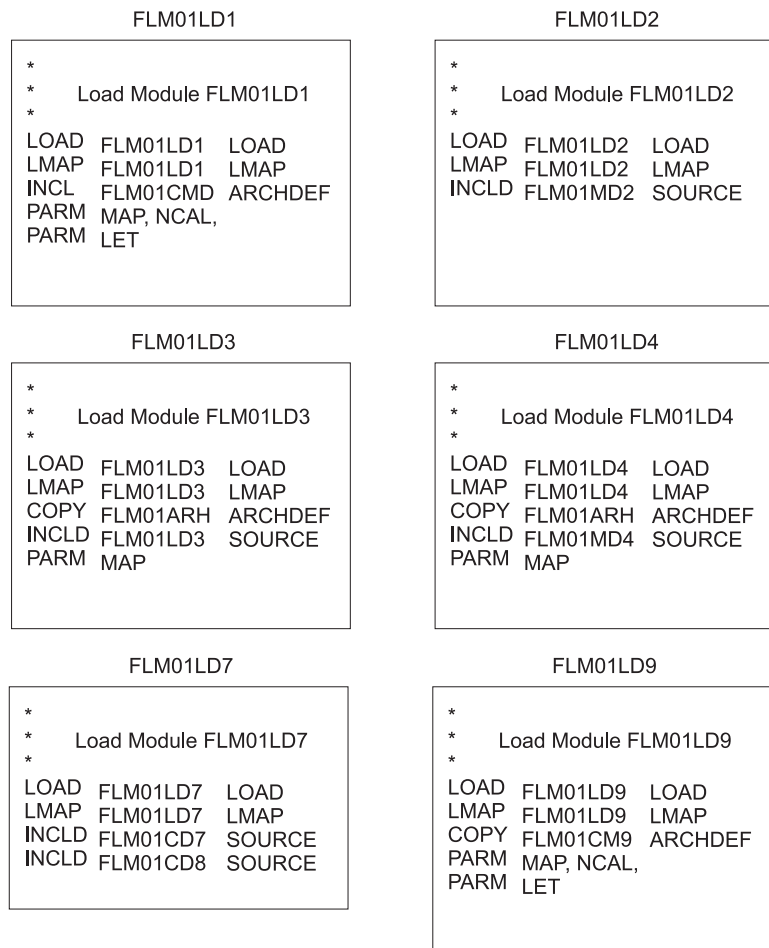
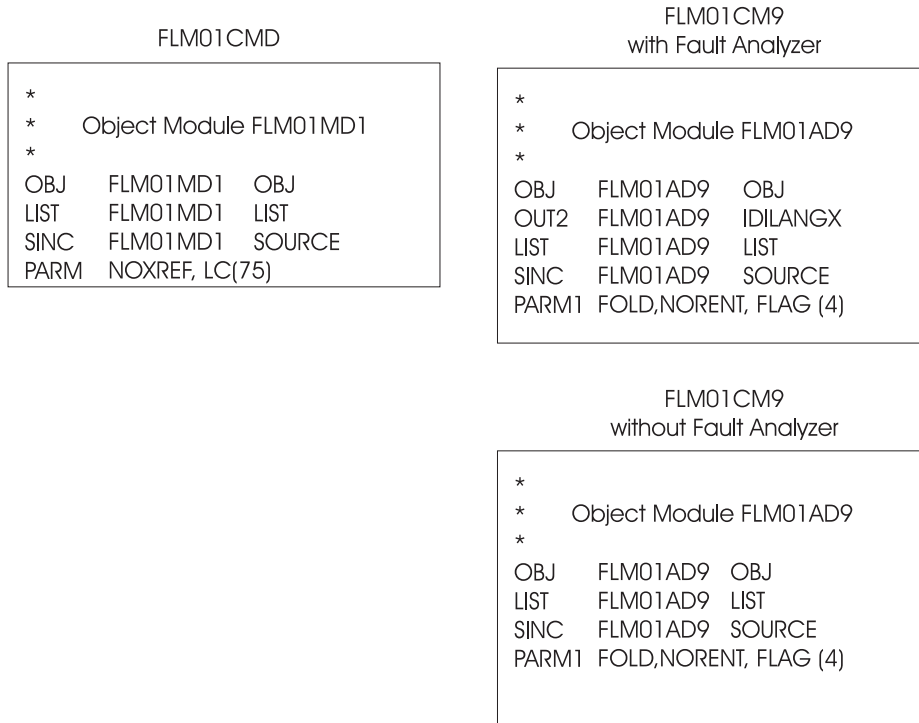


Figure 15. Architecture Members for Application Sample FLM01AP1 (Part 1 of 2)

Compilation Control Architecture Members



Copy Architecture Members

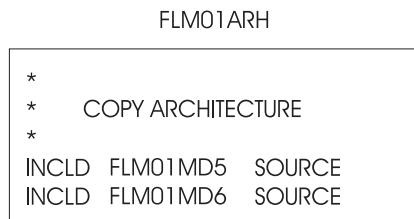


Figure 15. Architecture Members for Application Sample FLM01AP1 (Part 2 of 2)

Chapter 13, "SCLM support for DB2"

"Information for project administrators" section

"Generating a project environment" subsection: In **Step 8: Create the project definition**, add the following entries to Table 20, Language definitions for DB2:

FLM@2CBE	COB3DB2	Enterprise COBOL with DB2
FLM@2CBF	COB3DB2F	Enterprise COBOL with DB2 and Fault Analyzer
FLM@2CCE	CB3C2	Enterprise COBOL with DB2 and CICS
FLM@2PLE	PLEDB2	Enterprise PL/I with DB2
FLM@2PLF	PLEDB2F	DB2 and PL/I enterprise compiler and NCAL linkedit to a sub-module library with Fault Analyzer side file generation.

Chapter 18, "SCLM macros"

"FLMALLOC MACRO" section

"Parameters" subsection:

Figure 158 (Sample language definition for Assembler)

Replace:

```
* OS/VS ASSEMBLER LANGUAGE DEFINITION FOR SCLM
```

with:

```
*HIGH LEVEL ASSEMBLER LANGUAGE DEFINITION FOR SCLM
```

Replace:

```
COMPILE=IFOX00
```

with:

```
COMPILE=ASMA90
```

Delete:

```
FLMALLOC IOTYPE=W,DDNAME=SYSUT2,RECNUM=15000
```

```
FLMALLOC IOTYPE=W,DDNAME=SYSUT3,RECNUM=15000
```

Replace:

```
FLMALLOC IOTYPE=0,DDNAME=SYSGO,KEYREF=OBJ,RECNUM=7500,DFLTYP=OBJ
```

with:

```
FLMALLOC IOTYPE=0,DDNAME=SYSLIN,KEYREF=OBJ,RECNUM=7500,DFLTYP=OBJ
```

Replace:

```
FLMALLOC IOTYPE=0,DDNAME=SYSPRINT,KEYREF=LIST,PRINT=Y,          C  
DFLTYP=SOURCLST,RECNUM=20000
```

with:

```
FLMALLOC IOTYPE=0,DDNAME=SYSPRINT,KEYREF=LIST,PRINT=Y,          C  
DFLTYP=LIST,RECNUM=20000
```

Replace:

```
* 5665-402 (C) COPYRIGHT IBM CORP 1980, 1989
```

with:

```
* 5694-A01      COPYRIGHT IBM CORP 1980, 2007
```

IOTYPE=I

In the section, **Example:**, replace the existing statements with:

```
*  
* Object Module 1  
* OBJ FLM01MD1 OBJ  
LIST FLM01MD1 LIST  
SINC FLM01MD1 SOURCE  
PARM NOXREF,LC(75)
```

IOTYPE=O

In the section, **Example:**, replace:

```
"...LIST card, FLM01MD1 SOURCLST."
```

with:

"...LIST card, FLM01MD1 LIST."

IOTYPE=P

In the section, **Example**, replace the existing paragraph (starting "Figure 158 on page 458 does not contain...") with:

Figure 158 on page 458 does not contain an example of IOTYPE=P. However, if the Assembler, ASMA90, had required a partitioned data set for the object module instead of a sequential data set, then the FLMALLOC for the SYSLIN DDNAME would have used IOTYPE=P instead of IOTYPE=O.

IOTYPE=U

- In the section, **Example**, replace:

"...page 465, the assembler compiler, IFOX00, would want to use the output from the preprocessor, IFPRE0. It would not be necessary for IFOX00 to create a data set that would contain the input stream because this has been prepared by IFPRE0."

with:

"...page 465, the assembler compiler, ASMA90, would want to use the output from the preprocessor, IFPRE0. It would not be necessary for ASMA90 to create a data set that would contain the input stream because this has been prepared by IFPRE0."

- In figure 159 (Sample language definition that Calls a Preprocessor):

– Replace:

```
COMPILE=IFOX00
```

with:

```
COMPILE=ASMA90
```

– Replace

```
FLMALLOC IOTYPE=U,DDNAME=SYSIN
FLMALLOC IOTYPE=W,DDNAME=SYSUT1,RECNUM=17500
FLMALLOC IOTYPE=W,DDNAME=SYSUT2,RECNUM=15000
FLMALLOC IOTYPE=W,DDNAME=SYSUT3,RECNUM=15000
FLMALLOC IOTYPE=O,DDNAME=SYSGO,KEYREF=OBJ,RECNUM=7500,DFLTYP=OBJ
FLMALLOC IOTYPE=A,DDNAME=SYSTEM
  FLMCPYLB NULLFILE
FLMALLOC IOTYPE=A,DDNAME=SYSPUNCH
  FLMCPYLB NULLFILE
  FLMALLOC IOTYPE=I,DDNAME=SYSLIB,KEYREF=SINC
* ADD ONE FLMCPYLB FOR EACH FLMSYSLB
FLMCPYLB SYS1.MACLIB
FLMALLOC IOTYPE=O,DDNAME=SYSPRINT,KEYREF=LIST,PRINT=Y,          C
DFLTYP=SOURCLST,RECNUM=20000
```

with:

```
FLMALLOC IOTYPE=O,DDNAME=SYSLIN,KEYREF=OBJ,RECNUM=7500,DFLTYP=OBJ
FLMALLOC IOTYPE=N
FLMALLOC IOTYPE=N
FLMALLOC IOTYPE=I,DDNAME=SYSLIB,KEYREF=SINC
* ADD ONE FLMCPYLB FOR EACH FLMSYSLB
FLMCPYLB SYS1.MACLIB
FLMALLOC IOTYPE=U,DDNAME=SYSIN
FLMALLOC IOTYPE=O,DDNAME=SYSPRINT,KEYREF=LIST,PRINT=Y,          C
DFLTYP=LIST,RECNUM=20000
FLMALLOC IOTYPE=A,DDNAME=SYSPUNCH
  FLMCPYLB NULLFILE
FLMALLOC IOTYPE=W,DDNAME=SYSUT1,RECNUM=15000
FLMALLOC IOTYPE=N
```



```

FLMALLOC IOTYPE=N
FLMALLOC IOTYPE=N
FLMALLOC IOTYPE=A,DDNAME=SYSTEM
FLMCPYLB NULLFILE

```

IOTYPE=W

In the section, **Example:**, replace:

"Figure 158 on page 458 uses IOTYPE=W to allocate SYSUT1. When the Assembler, IFOX00, is invoked, a sequential data set is created and allocated to DDNAME SYSUT1. This data set is used..."

with:

"Figure 158 on page 458 uses IOTYPE=W to allocate SYSUT1. When the Assembler, ASMA90, is invoked, a sequential data set is created and allocated to DDNAME SYSUT1. This data set is used..."

UA36014, UA36871, UA32895

Release Date: 28 September 2007

The changes in this section apply to the following APARs and PTFs:

z/OS release	PTF #	APAR Abstract	APAR #	Doc Impact
1.8	UA32895 UA37047	Leave copybook behind (ICOPY).	OA18614	V1R8 Software Configuration and Library Manager (SCLM) Guide and Reference (SC34-4817-06).
1.7	UA36871 UA36717		OA19915	
1.6	UA36014 UA37074 UA36009		OA19916	

Chapter 9, "Using SCLM functions", subsection "Utilities (option 3)"

"Library utility" section

SCLM Library Utility - Entry Panel (figure 54)

This panel has been updated:

```

Menu  SCLM  Utilities  Help
-----
                                SCLM Library Utility - Entry Panel
Option ==> _____

blank Display member list          V View member
  A Browse accounting record       C Build member
  M Browse build map               P Promote member
  B Browse member                  U Update authorization code
  D Delete member, acct, bmap      T Transfer ownership
  E Edit member                    N Change promote processing

SCLM Library:
Project . . : SLMTEST6
Group . . . DEV1
Type . . . . SOURCE
Member . . . _____ (Blank or pattern for member selection list)

Select and rank member list data . . TAM (T=TEXT, A=ACCT, M=BMAP)

Enter "/" to select option
/ Hierarchy view                    Process . . 3 1. Execute
7 Confirm delete                     2. Submit
7 View processing options for Edit   3. View options
_ Show Member Description
F1=Help    F2=Split    F3=Exit    F7=Backward F8=Forward F9=Swap
F10=Actions F12=Cancel

```

Library Utility commands

Add a new command, N:

- N SCLM modifies the "account status" on the accounting record to determine if the editable member is promotable or not. This option is only available on members with an accounting record and an "accounting status" of:
 - EDITABLE
 - NOPROM-N, or
 - NOPROM-R

For more information on this option, see "New chapter ("Leaving a member behind on promotion")" on page 54.

Member selection list

- The Member Selection List (FLMUSL#P) panel shown in Figure 55 has been updated:

```

Menu  SCLM  Functions  Utilities  Help
-----
Member List : SLMTEST6.DEV1.SOURCE                               Member 1 of 23
Command ==> _____ Scroll ==> CSR

A=Account  M=Map      B=Browse   D=Delete   E=Edit     V=View
C=Build    P=Promote  U=Update   T=Transfer N=Noprom

Member  Status  Text      Chg Date  Chg Time  Account  Bld Map
-----
AAAA      DEV1      2002/08/02 13:31:12  DEV1      TEST
-
CPYRITE   DEV1      2002/01/21 13:08:15  DEV1
-
DDDDD     DEV1      2002/06/27 10:43:30
-
DTL2      DEV1      2002/01/21 13:08:04  DEV1      DEV1
-
FLM01EQU DEV1      2002/04/11 09:43:53  DEV1
-
FLM01MD1 DEV1      2002/02/14 12:24:05  DEV1      DEV1
-
FLM01MD2 DEV1      2002/02/14 12:24:10  DEV1      DEV1
-
FLM01MD3 DEV1      2002/02/14 12:23:52  DEV1      DEV1
-
FLM01MD4 RELEASE  2001/10/30 16:58:57  RELEASE   RELEASE
-
FLM01MD5 RELEASE  2001/10/30 16:58:57  RELEASE   RELEASE
-
FLM01MD6 DEV1      2002/01/22 13:06:08  DEV1      RELEASE
-
HANK      DEV1      2002/05/24 10:26:00  DEV1      DEV1
-
F1=Help   F2=Split  F3=Exit    F7=Backward F8=Forward F9=Swap
F10=Actions F12=Cancel

```

- The Member Selection List with Hierarchy and Member Description View (FLMUSM#P) panel shown in Figure 56 has been updated:

```

Menu  SCLM  Functions  Utilities  Help
-----
Member List : SLMTEST6.DEV1.SOURCE - HIERARCHY VIEW -           Member 1 of 23
Command ==> _____ Scroll ==> CSR

A=Account  M=Map      B=Browse   D=Delete   E=Edit     V=View
C=Build    P=Promote  U=Update   T=Transfer N=Noprom

Member  Status  Text      Chg Date  Chg Time  Account  Bld Map
-----
AAAA      DEV1      2002/08/02 13:31:12  DEV1      TEST
-
CPYRITE   DEV1      2002/01/21 13:08:15  DEV1
-
copywrite copy book
DDDDD     DEV1      2002/06/27 10:43:30
-
DTL2      DEV1      2002/01/21 13:08:04  DEV1      DEV1
-
DTL source for panel TTMENU
FLM01EQU DEV1      2002/04/11 09:43:53  DEV1
-
Assembler copybook - Register equates
FLM01MD1 DEV1      2002/02/14 12:24:05  DEV1      DEV1
-
F1=Help   F2=Split  F3=Exit    F7=Backward F8=Forward F9=Swap
F10=Actions F12=Cancel

```

- In the description of the fields that appear on the SCLM Member Selection List panel, change the description for the Status field to include the following value:

N Change Promote Processing *NOPROM

Accounting record

Change the description of the Accounting Status field to include:

NOPROM-N

Members that you can edit which, on promotion, are not copied to the next level. The build maps containing this member are not rebuilt, even if the associated language specifies an FLMLRBLD macro which would cause a rebuild.

NOPROM-R

Members that you can edit. On promotion the member and the build maps containing this member are not copied to the next level. Once the copy phase is complete, SCLM invokes a build to rebuild these build maps using the version of the member at the next level or above.

New subsection

Add the following as the last entry in the Library Utility section:

Change Promote Processing (NOPROM)

Type N on the Library Utility panel or against the member selection list to display the SCLM Not Promoted Member Update panel (FLMUSN#P).

```
Menu  SCLM  Utilities  Help
-----
                                SCLM Not Promoted Member Update
Command ==>> _____

SCLM Library:
PROJECT . . . . . : SLMTEST6
GROUP   . . . . . : DEV1
TYPE    . . . . . : SOURCE
MEMBER  . . . . . : FLM01EQU

Options
NOPROM: 1  1. No promote (Rebuild)
          2. No promote (No Rebuild)
          3. Remove no promote status

F1=Help    F2=Split    F3=Exit    F7=Backward  F8=Forward
F9=Swap    F10=Actions  F12=Cancel
```

Figure 16. SCLM Not Promoted Member Update panel (FLMUSN#P)

The fields on this panel are:

PROJECT

The SCLM project currently being maintained.

GROUP

Group of the member that is being changed.

TYPE Type of the member that is being changed.

MEMBER

Member that that is being changed.

NOPROM

Determines how the promote processing for this member is to be modified. Valid values are:

No promote (No Rebuild)

Sets the Accounting Status field in the accounting record to NOPROM-N. On promotion, the member is not copied to

the next level. The build maps containing this member are not rebuilt, even if the associated language specifies an FLMLRBLD macro which would cause a rebuild.

No promote (Rebuild)

Sets the Accounting Status field in the accounting record to NOPROM-R. On promotion, the build maps containing this member and the member itself are not copied to the next level. Once the copy phase is complete, SCLM invokes a build to rebuild these build maps using the version of the member at the next level or above.

Remove no promote status

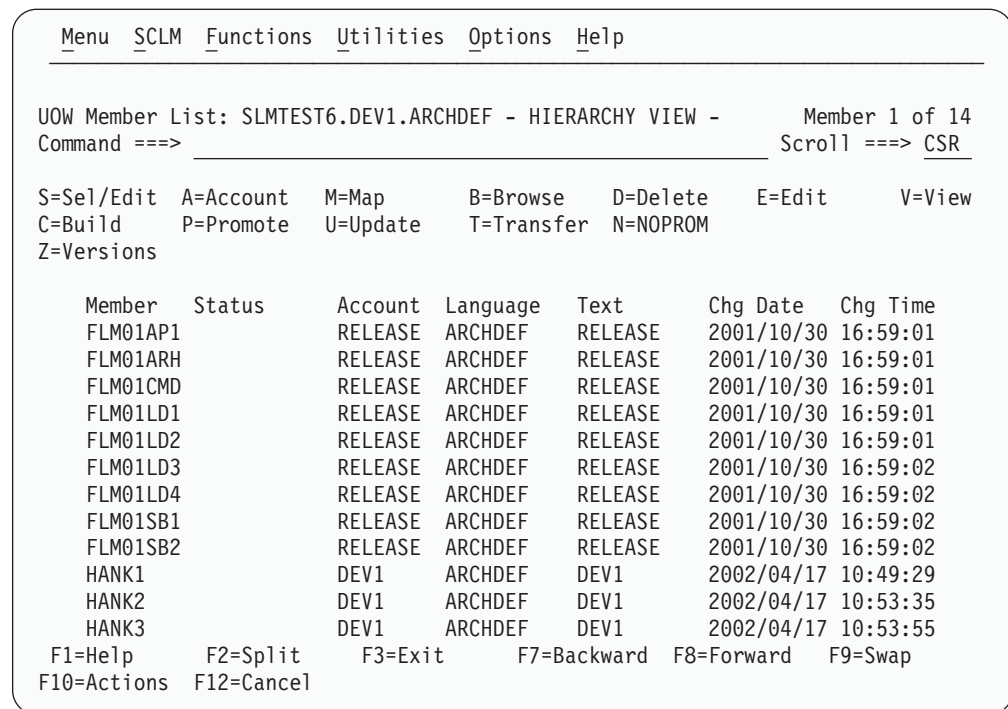
Sets the Accounting Status field in the accounting record to EDITABLE. Normal promotion of the member and associated build maps occurs.

For more information on this option, see the "New chapter ("Leaving a member behind on promotion")" on page 54.

"Unit of Work Utility" section

UOW Member List panel

- The UOW Member List panel shown in Figure 101 has been updated:



- In the description of the default commands available from the UOW Member List panel, add the following commands:

T=Transfer

SCLM modifies the Change User ID field in the accounting record to transfer ownership of the member to another user.

N=NOPROM

Allows you to modify how the SCLM member is promoted. You can use this option to leave the member behind on promotion.

Work Element List panel

- The Work Element List panel shown in Figure 102 has been updated:

```
Menu  SCLM  Functions  Utilities  Options  Help
-----
Work Element List for UOW FLM01AP1 in SLMTEST6                Member 1 of 2
Command ==> _____ Scroll ==> CSR
S=Sel/Edit  A=Account  M=Map      B=Browse   D=Delete   E=Edit     V=View
C=Build     P=Promote  U=Update   T=Transfer N=NOPROM
Z=Versions
  Member   Type      Status      Account   Language  Chg Date   User
  FLM01AP1 ARCHDEF   (Current   UOW ARCHDEF)
  FLM01SB1 ARCHDEF   RELEASE    ARCHDEF   2001/10/30 16:59 JPHILP
  FLM01SB2 ARCHDEF   RELEASE    ARCHDEF   2001/10/30 16:59 JPHILP
***** Bottom of data *****
F1=Help    F2=Split   F3=Exit    F7=Backward F8=Forward  F9=Swap
F10=Actions F12=Cancel
```

- Replace the list of default commands available from the Work Element List panel with the following sentence:
"The Commands available are the same as those available on the UOW Member List panel."

New chapter ("Leaving a member behind on promotion")

After Chapter 14 ("SCLM support for workstation builds"), add a new chapter entitled "Leaving a member behind on promotion" containing the following text.

Note: APAR OA18614 has introduced new functionality to allow a member to be left behind during promotion. Prior to this APAR being applied, SCLM copies all the required components to the next level.

You can specify whether an EDITABLE member and its accounting record are promoted to the next level or not. During promotion, the outputs (load modules) built using the non-promoted EDITABLE member are either:

- Rebuilt at the next level (REBUILD), or
- Not rebuilt at the next level

Sample scenarios for each of the above are described below.

Load modules rebuilt at the next level (REBUILD)

Scenario: There are DB2 table changes in the development group required for the updating of the DB2 DCLGEN copybooks, but these DB2 DCLGEN copybooks are not to be promoted until the DB2 changes are complete. At this stage, you create a fix for a program which uses one of the DB2 DCLGEN copybooks but does not require the other DB2 development changes. This fix needs to be promoted to production.

When building at the development level, the DB2 copybook is used so the program can be tested. But when promoting the program fix, you want the DB2 copybook not to be promoted and the program to be rebuilt at the next level using the version of the DB2 DCLGEN copybook at that level or above.

Load modules not rebuilt at the next level

Scenario: A copybook is being modified in development and a fix for a single program which uses the copybook needs to be promoted to production. However, promoting this copybook would cause problems when building other programs using the copybook after promotion.

You want to build using the development version of the copybook but, when promoting the program fix, wants the copybook not to be promoted and the program source, load and so on, promoted as is.

Note: By promoting everything except the copybook and its accounting record, the build map containing the non-promoted member is in a broken state. This is because the build map date and time for the non-promoted member does not match the member account record and member statistics date and time.

Setting a member as not being promotable

To specify that a member is non-promotable, use one of the following:

- The N line command in Library Utilities (option 3.1) or Unit of Work (option 3.11)
- FLMCMD NOPROM service
- FLMLNK NOPROM service

You can specify that an EDITABLE member is:

Non-promotable but the build maps containing the non-promoted member are rebuilt after promotion

For the N line command, you specify "No promote (Rebuild)" on the SCLM non-promoted Member Update panel or, if using the FLMCMD/FLMLNK NOPROM service, you specify the REBUILD parameter. This sets the Account Status field in the accounting record to NOPROM-R.

The accounting status of NOPROM-R is used during build and promote to tell SCLM not to promote the member and ensure the build maps containing the member are rebuilt.

Non-promotable and the build maps containing the non-promoted member are not rebuilt after promotion

For the N line command, you specify "No promote (No Rebuild)" on the SCLM Not Promoted Member Update panel or, if using the FLMCMD/FLMLNK NOPROM service, you specify the NOREBUILD parameter. This sets the Account Status field in the accounting record to NOPROM-N.

The accounting status of NOPROM-N is used during build and promote to tell SCLM not to promote the member and ensure the build maps containing the member are copied as is.

Promotable

For the N line command, you specify "Remove no promote status" on the SCLM Not Promoted Member Update panel or, if using the FLMCMD/FLMLNK NOPROM service, you specify the REMOVE parameter. This sets the Account Status field in the accounting record to EDITABLE.

You can only issue this option against a member with an Account Status of NOPROM-R or NOPROM-N.

Using the N line command in Library Utilities (option 3.1) or Unit of Work (option 3.11): You can only issue the N Line command in the Library Utilities (option 3.1) or Unit of Work (option 3.11) against a member with an accounting record that has an accounting status of EDITABLE, NOPROM-N, or NOPROM-R.

The N line command displays the SCLM Not Promoted Member Update panel.

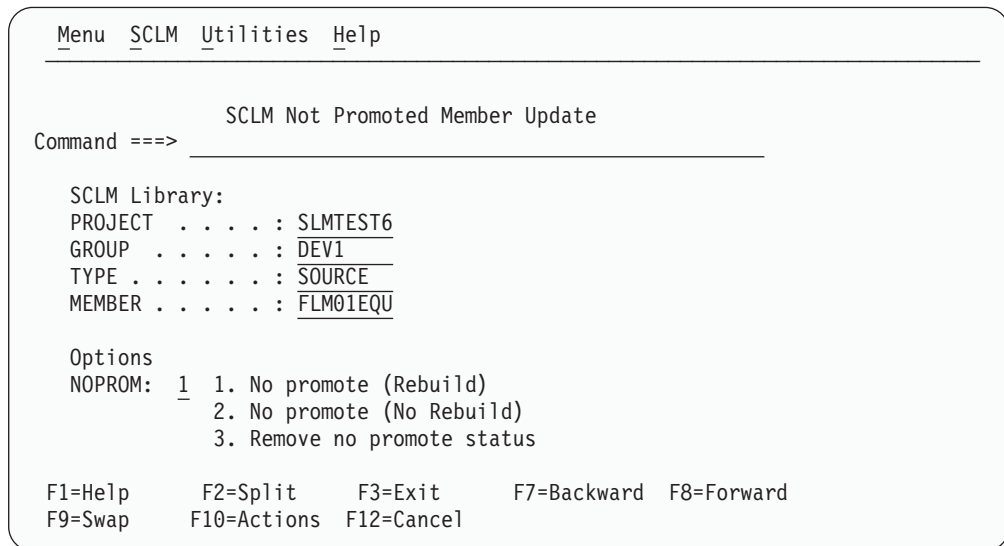


Figure 17. SCLM Not Promoted Member Update panel (FLMUSN#P)

When you press Enter, SCLM changes the Account Status field to one of the following:

- NOPROM-R, if you specify the "No promote (Rebuild)" option.
- NOPROM-N, if you specify the "No promote (No Rebuild)" option.
- EDITABLE, if you specify the "Remove no promote status" option *and* the account status is currently NOPROM-N or NOPROM-R.

FLMCMD NOPROM service: The following example shows invoking the FLMCMD NOPROM service to set the member FLM01EQU as non-promotable. Using the REBUILD parameter causes all the build maps containing FLM01EQU to be rebuilt when promoted.

For more information, see the NOPROM service in Chapter 15.

```

/*REXX*/
/*-----*/
/*
/* FLMCMD NOPROM command
/*
/*-----*/
trace n
/* allocate FLMSG DSN */
PARSE ARG PROJECT ALTPROJ GROUP TYPE MEMBER SVC Parm

ADDRESS TSO
msgdsn = ""||USERID()||".TEMP.FLMSG"
X=MSG('OFF')
IF SYSDSN(msgdsn) <> "OK" THEN DO
  "ALLOC FI(AAAMSGS) DATASET(msgdsn) ",
  "UNIT(SYSDA) TRACKS SPACE(5 5) ",
  "LRECL(80) BLKSIZE(3120) RECFM(F B) DSORG(PS) ",
  "NEW CATALOG "
  "FREE FI(AAATEMP)"

```



```

END
"FREE FI (FLMMSG) "
X=MSG('ON')
"ALLOC FI (FLMMSG) DATASET("msgdsn") SHR REUSE"

PROJECT="SCLMPROJ"
ALTPROJ="SCLMPROJ"
GROUP  ="DEV1"
TYPE   ="SOURCE"
MEMBER ="FLM01EQU"
SVC Parm="REBUILD"

"FLM CMD NOPROM,"PROJECT","ALTPROJ","GROUP","TYPE","MEMBER",",
", "SVC Parm",FLMMSG"

"ISPEXEC BROWSE DATASET("msgdsn")"
return

```

FLMLNK NOPROM service: The following example shows invoking the FLMLNK NOPROM service to set the member FLM01EQU as non-promotable. Using the NOREBUILD parameter causes SCLM to promote everything except the FLM01EQU member and its accounting record.

For more information, see the NOPROM service in Chapter 15.

```

IDENTIFICATION DIVISION.
PROGRAM-ID. FLM02CBL.
*@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
*@@.THIS PROGRAM ALLOWS YOU TO CALL SCLM SERVICES FROM A COBOL @
*@@ PROGRAM. @
*@@ @ @
*@@.THE FUNCTION OF THIS PROGRAM IS TO PERFORM AN SCLM NOPROM ON @
*@@ THE COPYBOOK FLM01EQU IN THE DEV1 GROUP. @
*@@ @ @
*@@ 5647-A01 (C) COPYRIGHT IBM CORP. 1987, 2000 @
*@@ @ @
*@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
01 PROGRAM-NAME PIC X(08) VALUE 'FLM02CBL'.
01 SCLM-SERVICE PIC X(08).
01 SCLM-PROJECT PIC X(08).
01 SCLM-ALT-PROJ PIC X(08).
01 SCLM-GROUP PIC X(08).
01 SCLM-TYPE PIC X(08).
01 SCLM-MEMBER PIC X(08).
77 SCLM-SCLM-ID PIC X(08) VALUE SPACE.
77 SCLM-APPL-ID PIC X(08) VALUE SPACE.
77 SCLM-AUTHCODE PIC X(08) VALUE SPACE.
77 SCLM-NOPROM-FIELD PIC X(16).
77 SCLM-ACCESS-KEY PIC X(16) VALUE SPACE.
77 SCLM-ACCT-INFO PIC S9(04) COMP VALUE ZERO.
77 SCLM-LIST-INFO PIC S9(04) COMP VALUE ZERO.
77 SCLM-MSG-ARRAY PIC S9(04) COMP VALUE ZERO.
77 SCLM-DD-PRSLIST PIC X(08) VALUE 'SYSOUT'.
77 SCLM-DD-MSG PIC X(08) VALUE 'MDS3602M'.
77 SCLM-DD-EXIT PIC X(08) VALUE SPACE.
77 SCLM-MSG-LINE PIC X(80) VALUE SPACE.
*
LINKAGE SECTION.
PROCEDURE DIVISION.
1-DRIVER.
*-----
* FLMLNK NOPROM,JPHILP,OS2G,DEV1,SOURCE,FLM01EQU,NOREBUILD
*-----

```

```

MOVE 'SCLMPROJ'      TO SCLM-PROJECT
MOVE 'SCLMPROJ'      TO SCLM-ALT-PROJ
MOVE 'DEV1'          TO SCLM-GROUP
MOVE 'SOURCE'        TO SCLM-TYPE
MOVE 'FLM01EQU'      TO SCLM-MEMBER
MOVE 'NOREBUILD'     TO SCLM-NOPROM-FIELD

MOVE 'START'         TO SCLM-SERVICE
CALL 'FLMLNK' USING SCLM-SERVICE
                    SCLM-APPL-ID.
DISPLAY 'START RETURN CODE = ' RETURN-CODE.
MOVE 'INIT'          TO SCLM-SERVICE
CALL 'FLMLNK' USING SCLM-SERVICE
                    SCLM-APPL-ID
                    SCLM-PROJECT
                    SCLM-ALT-PROJ
                    SCLM-SCLM-ID
                    SCLM-MSG-LINE.
DISPLAY 'INIT RETURN CODE = ' RETURN-CODE.

MOVE 'NOPROM'        TO SCLM-SERVICE
CALL 'FLMLNK' USING SCLM-SERVICE
                    SCLM-SCLM-ID
                    SCLM-GROUP
                    SCLM-TYPE
                    SCLM-MEMBER
                    SCLM-NOPROM-FIELD
                    SCLM-ACCESS-KEY
                    SCLM-MSG-LINE.
DISPLAY 'NOPROM RETURN CODE = ' RETURN-CODE.
IF RETURN-CODE > 0 THEN
    DISPLAY SCLM-MSG-LINE.
MOVE 'FREE'          TO SCLM-SERVICE
CALL 'FLMLNK' USING SCLM-SERVICE
                    SCLM-SCLM-ID
                    SCLM-MSG-LINE.
DISPLAY 'FREE RETURN CODE = ' RETURN-CODE.

MOVE 'END'           TO SCLM-SERVICE
CALL 'FLMLNK' USING SCLM-SERVICE
                    SCLM-APPL-ID
                    SCLM-MSG-LINE.
DISPLAY 'END RETURN CODE = ' RETURN-CODE.
DISPLAY ' '.
DISPLAY 'NOPROM COMPLETED'.
DISPLAY 'LOAD FLM02CBL ENDED'.

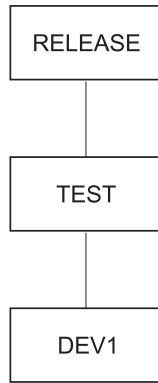
GOBACK.
1-EXIT. EXIT.

```

Process of not promoting a member (REBUILD)

This section describes the process of not promoting a member and causing a REBUILD of the load modules which use it at the next level.

For the following description of this project the project will have the following structure.



Scenario: The DEV1 group contains a DB2 DCLGEN copybook SCLMTB01 for a DB2 table which is in the process of being modified. You want to build and test in DEV1 using this copybook but do not want SCLMTB01 promoted until it is ready.

Using the new functionality the member can be modified so it is not copied to the next level during promotion. Any load modules which use SCLMTB01 are rebuilt after promotion to the next level (TEST) to use the version of the copybook at TEST or above.

To do this, you can use either the N line command in the Library Utility (option 3.1) or Unit of Work (option 3.11) or the FLMCMD/FLMLNK NOPROM service. For this example, we will use the N line command in the Library Utility option. By issuing the N line command, SCLM invokes the following panel.

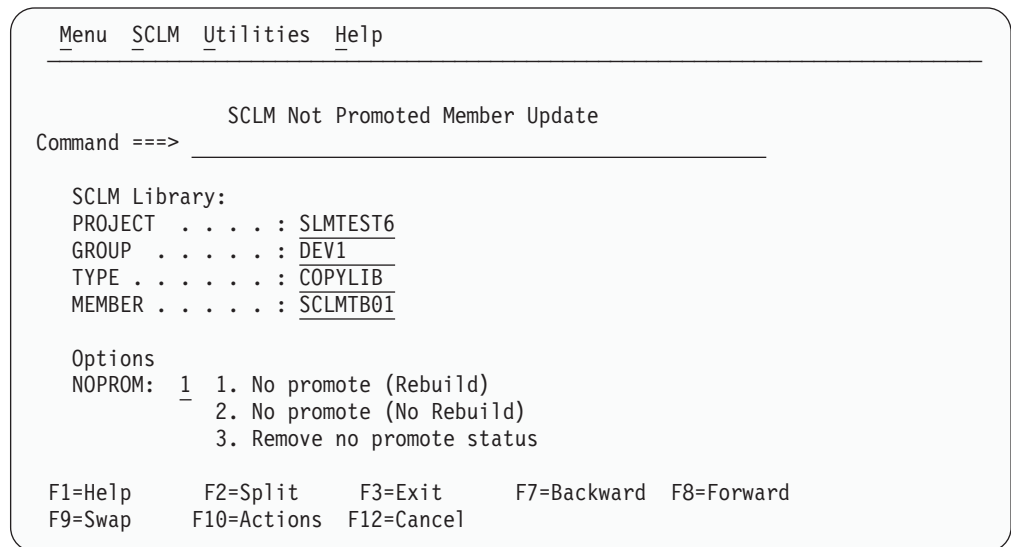


Figure 18. SCLM Not Promoted Member Update panel (FLMUSN#P)

By specifying "No promote (Rebuild)", SCLM modifies the accounting record for SCLMTB01 to have an account status of NOPROM-R.

```

SLMTEST6.DEV1.COPYLIB(SCLMTB01): Accounting Record
Command ==>>> _____

Physical Data Set . . : SLMTEST6.DEV1.COPYLIB
Accounting Status . . : NOPROM-R           Change Group . . . . : DEV1
Change User ID . . . : JPHILP             Authorization Code . . : P
Member Version . . . : 2                  Auth. Code Change . . :
Language . . . . . : TEXT                 Translator Version . . :
Creation Date . . . . : 2007/01/08        Change Date . . . . . : 2007/01/16
Creation Time . . . . : 16:55:08          Change Time . . . . . : 09:51:05
Promote User ID . . . :                    Access Key . . . . . :
Promote Date . . . . : 0000/00/00         Build Map Name . . . . :
Promote Time . . . . : 00:00:00           Build Map Type . . . . :
Predecessor Date . . : 2007/01/08        Build Map Date . . . . : 2007/01/16
Predecessor Time . . : 16:56:22          Build Map Time . . . . : 09:51:05

Enter "/" to select option
Display Statistics
Number of Change Codes      : 0
Number of Includes         : 0
Number of Includes         : 0
Number of Compilation Units : 0
Number of User Entries     : 0
F1=Help      F2=Split    F3=Exit      F7=Backward  F8=Forward   F9=Swap
F12=Cancel

```

While building the record, SCLM analyzes the components and determines that the member SCLMTB01 has an account status of NOPROM-R. Once the build has completed, SCLM updates the build maps containing the SCLMTB01 member to have a NOPROM build map record. The build report shows the build maps that were updated.

```

***** N O P R O M   M E M B E R S                ***** Page 5

BUILD   BMAP      NOPROM   NOPROM MBR
MAP     TYPE      MEMBER   TYPE
-----
SCLMDB2 SOURCE    SCLMTB01 COPYLIB

***** Bottom of Data *****

```

The build map for the SCLMDB2 COBOL member looks as follows.

```

SLMTEST6.DEV1.SOURCE(SCLMDB2): Build Map Contents
Command ==> _____ Line
                               Build Map Contents   Scroll ==> CSR
                               -----
Keyword  Member                               Type    Last Time Modified  Ver
-----
SINC     SCLMDB2                               SOURCE   2007/01/25 12:06:52 6
OBJ      SCLMDB2                               OBJ      2007/01/25 12:06:55 6
LIST     SCLMDB2                               LIST     2007/01/25 12:06:55 6
OUT1     SCLMDB2                               DBRM     2007/01/25 12:06:55 6
I1*      COBCOPY1                               COPYLIB  2007/01/08 16:57:59 1
NOPROM   SCLMTB01                               COPYLIB
I1*      SCLMTB01                               COPYLIB  2007/01/16 09:51:05 2
I1*      SCLMTB02                               COPYLIB  2007/01/16 09:51:12 3

* Internal Keywords
  I#      - Included member referenced by SINC member, # = Imbedded Group
  NOPROM - Member was/will be left behind on promotion.
***** Bottom of Data *****

F1=Help   F2=Split   F3=Exit   F4=retrieve F5=Rfind   F7=Up
F8=Down   F9=Swap    F10=Left  F11=Right  F12=Cancel

```

The NOPROM build map record is used during promote to indicate that the promotion contains members that are not to be copied to the next level.

When promoting SCLMDB2 from DEV1 to TEST, SCLM does the following:

- Encounters the NOPROM map record on the build map SCLMDB2.
- Compares the group specified on the NOPROM map record (DEV1) and, as it is same as the group being promoted from (DEV1), the normal date and time validation of the non-promoted member in the build maps takes place.
- Verifies that the SCLM components are current by comparing the build maps, accounting records, and member statistics.
- While performing this verification, SCLM checks the accounting record for the accounting status for the SCLMTB01 member. As it is set to NOPROM-R, SCLM does not copy to the next level the member SCLMTB01, its accounting record, or any of the build maps containing the NOPROM member SCLMTB01.
- When copying components to the next level, SCLM does not copy the SCLMTB01 member, its accounting record, or any of the build maps (that is, SCLMDB2) containing the SCLMTB01 member.
- Once the copy phase is complete, SCLM invokes a build at the TEST level. As the build maps containing the SCLMTB01 member were not promoted, these are rebuilt at the TEST level using the version of SCLMTB01 at the TEST level or above.

The promote report shows the non-promoted members and the build maps that were affected by the non-promoted member.

```

*****
**
**          N O T   P R O M O T E D   M E M B E R S
**
**
*****
BACKUP   NOT PROMOTED MEMBER   TYPE   MEMBER   OPTION   _____
-----
SCLMTB01 COPYLIB                REBUILD

                                           PAGE 10
*****
**
**          B U I L D   M A P S   A F F E C T E D   B Y   N O T   P R O M O T E D   M E M B E R S
**
**
*****
BUILD    BMAP    BMAP    NOPROM    NOPROM MBR MAP    TYPE    REBUILT
MEMBER   TYPE   _____
SCLMDB2  SOURCE  YES      SCLMTB01 COPYLIB

***** Bottom of Data *****

```

Once the promote is complete, when you view the SCLMDB2 build you can see the SCLMTB01 member used to build SCLMDB2 is dated 2007/01/08 16:56, not 2007/01/16 09:51. This shows that SCLMDB2 was built using an earlier version of the copybook SCLMTB01 at the TEST level, or higher.

```

SLMTEST6.TEST.SOURCE(SCLMDB2): Build Map Contents          Line
Command ==> _____ Scroll ==> CSR
                          Build Map Contents
                          -----
Keyword  Member                Type      Last Time Modified  Ver
-----
SINC     SCLMDB2                     SOURCE    2007/01/25 12:06:52  6
OBJ      SCLMDB2                     OBJ       2007/01/25 12:10:22  7
LIST     SCLMDB2                     LIST      2007/01/25 12:10:22  7
OUT1     SCLMDB2                     DBRM     2007/01/25 12:10:22  7
I1*     COBCOPY1                    COPYLIB  2007/01/08 16:57:59  1
I1*     SCLMTB01                    COPYLIB  2007/01/08 16:56:22  1
I1*     SCLMTB02                    COPYLIB  2007/01/16 09:51:12  3

* Internal Keywords
  I#     - Included member referenced by SINC member, # = Imbedded Group
***** Bottom of Data *****

F1=Help    F2=Split    F3=Exit    F4=retrieve  F5=Rfind    F7=Up
F8=Down    F9=Swap    F10=Left   F11=Right   F12=Cancel

```

Process of not promoting a member (NOREBUILD)

This section describes the process of not promoting a member but promoting the load modules and related members (NOREBUILD).

Note: Prior to this new functionality, promotion of SCLM members left the SCLM environment in a complete state where the dates and times in the

accounting records, member statistics, and the build maps matched.
Performing a build at this level does not result in any members being built.

The new functionality of promoting everything except a member and its accounting record (No Promote - No Rebuild) results in SCLM being in a broken state. The build maps containing the non-promoted member have dates and times for the member which was not promoted. This means that SCLM needs to cater for several issues when promoting and building using the FLMCMD/FLMLNK NOPROM service and specifying the NOREBUILD parameter or using the N line command and specifying "No promote (No Rebuild)" on the SCLM Not Promoted Member Update panel. These issues are:

- **Need to backup the non-promoted member.** Since you are promoting the build maps that were built using the non-promoted member, you need to take a backup of this member. This ensures that, even if the member has been modified or deleted after you promoted, you can still recreate the outputs (load modules).
- **SCLM needs to be able to promote build maps at a level where a member was left behind.** After promotion, where a member was left behind but the build maps were not rebuilt, the level you promoted into is in a broken state. Normally, when attempting to promote these changes further, SCLM gave you date and time inconsistencies between the build map and accounting record for the non-promoted members.

For the new functionality, SCLM has been changed so that when promoting it ignores the date and time inconsistencies between the build map, accounting record, and member statistics if:

- The build map contains a NOPROM build map record, and
- The build level (not viewable) found on the NOPROM build map record does not match the current promote from level

- **SCLM needs to be able to build at a level where a member was left behind.** After promotion, where a member was left behind but the build maps were not rebuilt, the level you promoted into is in a broken state. Normally, when attempting to build at this level, it causes the build maps with date and time inconsistencies (due to left-behind members) to be rebuilt.

For the new functionality, SCLM has been changed so that when building it ignores the date and time inconsistencies between the build map, accounting record, and member statistics if:

- The build map contains a NOPROM build map record, and
- The build level (not viewable) found on the NOPROM build map record does not match the current promote from level

SCLM project setup when promoting with no rebuilding of build maps: To allow SCLM to backup the non-promoted members, you must perform the following:

- Allocate a NOPROM backup data set. This data set is a fixed block data set with a LRECL of 1024.
- Modify the SCLM project definition to add a NPROMBK parameter on the FLMCNTRL macro to specify the NOPROM backup data set name.
- Allocate a CONTROL VSAM data set (if not already defined). The sample JCL for defining this member can be located in ISP.SISPSAMP(FLM02CNT).
- Modify the SCLM project definition to add in a CONTROL parameter on the FLMCNTRL macro to specify the CONTROL VSAM data set name.

If the above steps have not been performed, SCLM still completes a promotion containing the non-promoted member; however, no backups are taken. To recreate

the outputs (for example, the Load module) based on the left-behind member, you must use the left-behind member. If this member was modified or deleted after promotion, it may not be possible to recreate the SCLM outputs.

Build containing a non-promotable member (NOREBUILD): The following is based on the SCLM sample project.

Scenario: A copybook FLM01EQU is being modified in development and a fix for a load module FLM01LD4 which uses the copybook needs to be promoted to production. However promoting the copybook FLM01EQU would cause problems when building other load modules (for example, FLM01LD3) using the copybook after promotion.

You therefore want to build using the development version of the copybook FLM01EQU, but when promoting the load module FLM01LD4 you do not want the copybook copied to the next level (TEST).

To set up FLM01EQU so that it is not promoted and that the load module FLM01LD4 is not rebuilt, you can use either the FLMCMD/FLMLNK NOPROM service or the N line command in the Library Utility (option 3.1) or Unit of Work (option 3.11). For this example, we will use the N line command in the Library Utility. By issuing the N line command, SCLM invokes the following panel.

```
Menu  SCLM  Utilities  Help
-----
SCLM Not Promoted Member Update
Command ==> _____

SCLM Library:
PROJECT . . . . . : SLMTEST6
GROUP   . . . . . : DEV1
TYPE    . . . . . : SOURCE
MEMBER  . . . . . : FLM01EQU

Options
NOPROM:  2  1. No promote (Rebuild)
           2. No promote (No Rebuild)
           3. Remove no promote status

F1=Help   F2=Split   F3=Exit    F7=Backward  F8=Forward
F9=Swap   F10=Actions  F12=Cancel
```

By specifying "No promote (No Rebuild)", SCLM modifies the accounting record for FLM01EQU to have an account status of NOPROM-N.


```

SLMTEST6.DEV1.SOURCE(FLM01EQU): Accounting Record
Command ==> _____

Physical Data Set . . : SLMTEST6.DEV1.SOURCE
Accounting Status . . : NOPROM-N           Change Group . . . . : DEV1
Change User ID . . . : JPHILP             Authorization Code . . : P
Member Version . . . : 2                   Auth. Code Change . . :
Language . . . . . : HLAS                 Translator Version . . :
Creation Date . . . . : 2001/10/30        Change Date . . . . . : 2007/01/25
Creation Time . . . . : 16:58:56          Change Time . . . . . : 12:14:38
Promote User ID . . . :                    Access Key . . . . . :
Promote Date . . . . : 0000/00/00         Build Map Name . . . . :
Promote Time . . . . : 00:00:00           Build Map Type . . . . :
Predecessor Date . . : 2001/10/30        Build Map Date . . . . : 2007/01/25
Predecessor Time . . : 16:58:56          Build Map Time . . . . : 12:14:38

Enter "/" to select option
Display Statistics
Number of Change Codes      : 1
Number of Includes         : 0
Number of Compilation Units : 0
Number of User Entries     : 1
F1=Help      F2=Split    F3=Exit      F7=Backward  F8=Forward   F9=Swap
F12=Cancel

```

While building the record, SCLM analyzes the components to be built and determines that the member FLM01EQU has an account status of NOPROM-N. Once the build has completed for FLM01LD4, SCLM updates the build maps containing the FLM01EQU member to have a NOPROM build map record.

The build report shows the build maps that were updated.

```

***** N O P R O M   M E M B E R S                ***** Page 5

BUILD   BMAP      NOPROM   NOPROM MBR
MAP     TYPE      MEMBER   TYPE
-----
FLM01MD4 SOURCE    FLM01EQU SOURCE
FLM01MD5 SOURCE    FLM01EQU SOURCE
FLM01MD6 SOURCE    FLM01EQU SOURCE

***** Bottom of Data *****

```

The build map for the FLM01MD4 member looks as follows.

```

Work Element List for UOW FLM01AP1 in SLMTEST6           Member 1 of 2
Command ==> _____                               Scroll ==> CSR

S=Sel/Edit  A=Account  M=Map      B=Browse   D=Delete   E=Edit     V=View
C=Build     P=Promote   U=Update   T=Transfer N=NOPROM
Z=Versions

Member  Type      Status      Account Language Chg Date      User
FLM01AP1 ARCHDEF          (Current UOW ARCHDEF)
FLM01SB1 ARCHDEF          RELEASE ARCHDEF  2001/10/30 16:59 JPHILP
FLM01SB2 ARCHDEF          RELEASE ARCHDEF  2001/10/30 16:59 JPHILP

***** Bottom of data *****

```

The NOPROM build map record in the build maps for FLM01MD4, FLM01MD5, and FLM01MD6 is used during promote to indicate that the promotion contains members that are not to be promoted.

Promote containing a non-promotable member (NOREBUILD) from the same level containing the NOPROM member: After the member has been specified as being non-promotable (NOREBUILD) and SCLM has performed a build, these changes can then be promoted to the next level (TEST).

When promoting FLM01LD4 from DEV1 to TEST, SCLM does the following:

- Encounters the NOPROM map record on the build maps FLM01MD4, FLM01MD5, and FML01MD6.
- Compares the group specified on the NOPROM map record (DEV1) and, as it is same as the group we are promoting from (DEV1), the normal date and time validation of the non-promoted member in the build maps takes place.
- Verifies that the SCLM components are current by comparing the build maps, accounting records, and member statistics.
- While performing this verification, SCLM checks the accounting record for the accounting status for the FLM01EQU member. As it is set to NOPROM-N, all the build maps containing the NOPROM member FLM01EQU are copied to the next level.
- Reads the CONTROL file to determine the name of the backup member. To do this, SCLM reads the CONTROL file to determine the NOPROM backup number; this is used to generate the backup member name. If the backup number is 00000001, then the backup member is A0000001.
- Backs up the non-promoted member FLM01EQU into the NOPROM PDS specified by the NPROMBK parameter on the FLMCNTRL macro in the project definition.
- Updates the build maps for FLM01MD4, FLM01MD5, and FML01MD6 at the DEV1 level to add in the backup member name generated above (A0000001) into the NOPROM build map record.
- When copying components to the next level, the FLM01EQU member and its accounting record are not promoted, but build maps containing the FLM01EQU member are promoted.

Viewing the non-promoted backup member: A build map containing NOPROM build map for which a backup has been taken looks as follows.

```

SLMTEST6.DEV1.SOURCE(FLM01MD4): Build Map Contents      Line
Command ==> _____ Scroll ==> CSR
                          Build Map Contents
                          -----
Keyword  Member                Type      Last Time Modified  Ver
-----
SINC     FLM01MD4                 SOURCE    2001/10/30 16:58:57  1
OBJ      FLM01MD4                 OBJ       2007/01/25 12:19:00  2
LIST     FLM01MD4                 SOURCLST 2007/01/25 12:19:00  2
NOPROM   FLM01EQU                 SOURCE
I1*      FLM01EQU                 SOURCE    2007/01/25 12:14:38  2

* Internal Keywords
I#       - Included member referenced by SINC member, # = Imbedded Group
NOPROM   - Member was/will be left behind on promotion.
          Use the S line command to view the not promoted member.

***** Bottom of Data *****

```

By typing an S alongside the NOPROM member, you can view the backup of the member that was taken as a part of the promotion of this build map.

Promote containing a non-promotable member (NOREBUILD) from a level not containing the NOPROM member: Scenario: You have tested the changes at TEST and want to promote the changes to the RELEASE level. However the SCLM components at TEST are in a broken state as build maps FLM01MD4, FLM01MD5, and FLM01MD6 at TEST contain the member FLM01EQU, but this member was not promoted from DEV1.

When promoting from TEST to RELEASE, SCLM does the following:

- Encounters the NOPROM map record in the build maps build maps FLM01MD4, FLM01MD5, and FLM01MD6. The FLM01EQU member is added to the list of members that are not being promoted.
- Compares the group specified on the NOPROM map record (DEV1) and, as it is not equal to the group being promoted from (TEST):
 - Skips the date and time validation of the non-promoted member, FLM01EQU
 - Does not backup the non-promoted member, FLM01EQU
- Verifies that the SCLM components are current by comparing the build maps, accounting records, and member statistics. If SCLM encounters the member FLM01EQU, it is flagged as being up to date.
- When copying components to the next level, the FLM01EQU member and its accounting record are not promoted, but build maps containing the FLM01EQU member are promoted.

Build containing a non-promotable member (NOREBUILD) at a level which does not contain the NOPROM member: After the changes have been promoted to RELEASE, the SCLM components are in a broken state as build maps FLM01MD4, FLM01MD5, and FLM01MD6 contain the member FLM01EQU, but this member was not promoted from DEV1.

Note: Prior to this new functionality, a build perform against FLM01LD4 rebuilt the load module due to the broken state of the build maps FLM01MD4, FLM01MD5, and FLM01MD6. This is not acceptable as it would override the changes you have just promoted.

When performing a build containing a NOPROM build map record where the level specified on the build maps record does not match the build level, SCLM performs the following:

- When analyzing the SCLM components, encounters the NOPROM map record on the build maps FLM01MD4, FLM01MD5 and FLM01MD6.

For each of these build maps, SCLM performs the following:

- Compares the group specified on the NOPROM map record (DEV1) to see if it is different to the group you are building at. If it is different, and the date and time of the accounting record does not match the date and time in the build maps, then SCLM knows that the member FLM01EQU has not been promoted.
- As the FLM01EQU member has not been promoted, it is set as "up to date" in the build map and SCLM issues message FLM44523.
- If other changes are encountered which would cause the build map to be rebuilt, SCLM issues message FLM44522 and completes with a return code of 8.

The above situation occurs if a member used by the build map was changed and promoted but the build map was not rebuilt. To resolve this problem, you need to do one of the following:

- Delete the build map causing the problem and rebuild. This causes the build map to be rebuilt using the version of the non-promoted member at the build level or above (that is, RELEASE).
- Rebuild the build map at the group containing the non-promoted member (DEV1) and re-promote your changes. This rebuilds the build map using the members at the higher level, as well as the non-promoted member.

Build after promotion of the non-promotable member (NOREBUILD): If the member FLM01EQU is ready to be promoted into RELEASE, to remove the non-promotable status of the member FLM01EQU, you can use the N line command in Library Utilities (option 3.1) or Unit of Work (Option 3.11), or run the FLMCMD/FLMLNK NOPROM service. Once this is complete, the accounting record for the member FLM01EQU looks as follows.

```

SLMTEST6.DEV1.SOURCE(FLM01EQU): Accounting Record
Command ==>>
-----
Physical Data Set . . : SLMTEST6.DEV1.SOURCE
Accounting Status . . : EDITABLE           Change Group . . . . : DEV1
Change User ID . . . : JPHILP             Authorization Code . . : P
Member Version . . . : 2                  Auth. Code Change . . :
Language . . . . . : HLAS                 Translator Version . . :
Creation Date . . . . : 2001/10/30        Change Date . . . . . : 2007/01/25
Creation Time . . . . : 16:58:56         Change Time . . . . . : 12:14:38
Promote User ID . . . :                   Access Key . . . . . :
Promote Date . . . . : 0000/00/00        Build Map Name . . . . :
Promote Time . . . . : 00:00:00         Build Map Type . . . . :
Predecessor Date . . : 2001/10/30       Build Map Date . . . . : 2007/01/25
Predecessor Time . . : 16:58:56        Build Map Time . . . . : 12:14:38

Enter "/" to select option
Display Statistics
Number of Change Codes      : 1
Number of Includes         : 0
Number of Compilation Units : 0
Number of Compilation Units : 0
Number of User Entries     : 0
F1=Help      F2=Split   F3=Exit      F7=Backward  F8=Forward   F9=Swap
F12=Cancel

```

By promoting the member FLM01EQU, or a build map containing FLM01EQU, SCLM invokes a promotion process that copies the member FLM01EQU and its accounting record to the next level. Once the member FLM01EQU has been promoted to the RELEASE level, if a rebuild of the build maps FLM01MD4, FLM01MD5, and FLM01MD6 has not occurred, these build maps still contain the NOPROM build map records.

When building at the RELEASE level after promotion of FLM01EQU, SCLM does the following:

- Encounters the NOPROM map record on the build maps FLM01MD4, FLM01MD5, and FLM01MD6.
- Compares the group specified on the NOPROM map record (DEV1) to see if it is different to the level at which you are building. If it is different, and the date and time of the accounting record matches the date and time in the build maps, then SCLM knows that the member FLM01EQU has been promoted.
- Removes the NOPROM build map record entries from the build maps FLM01MD4, FLM01MD5, and FLM01MD6, even if they were not re-built.

Restricting the setting of non-promotable: You can restrict the use of the N line command in the Library Utility (option 3.1) or Unit of Work (option 3.11), or the FLMCMD/FLMLNK NOPROM service, by the use of the FLMNPROM macro within the SCLM project definition.

If FLMNPROM has not been specified in the SCLM project definition, there are no restrictions on the use of the NOPROM service or the N line command in Library Utility (option 3.1) or Unit of Work (Option 3.11) within the SCLM project.

By specifying the FLMNPROM macro, you can specify which groups, types, and languages can be set as non-promotable.

Examples: The following example shows how to specify that editable members with a language of COBCOPY in the SOURCE type, in any group, can be marked as non-promotable:

```
FLMNPROM GROUP=*,TYPE=SOURCE,LANG=COBCOPY,NPROM=YES
```

The following example shows how to specify that all editable members with a language of COBCOPY in the SOURCE type, in all groups except EMERFIX, can be marked as non-promotable:

```
FLMNPROM GROUP=*,TYPE=SOURCE,LANG=COBCOPY,NPROM=YES  
FLMNPROM GROUP=EMERFIX,TYPE=SOURCE,LANG=COBCOPY,NPROM=NO
```

Chapter 15, "Invoking the SCLM services"

"Types of parameters" section

In the subsection, "Pointer parameter descriptions", add the following values to the **Contents** column for the "accounting_status" field:

X NOPROM-R (REBUILD)

Y NOPROM-N (NOREBUILD)

"ISPF variables" section

For the ZSASTAT variable, replace the existing description with the following:

"Status of the accounting record. Possible values are: EDITABLE, NON-EDIT, LOCKOUT, INITIAL, ERROR, NOPROM-N, and NOPROM-R."

Chapter 16, "SCLM services"

Add a new SCLM service, NOPROM. To do this:

- Add a new entry to the summary list at the start of the chapter (between NEXTGRP and PARSE):
"NOPROM—Change Promote processing"
- Add a new section describing NOPROM (between the descriptions for NEXTGRP and PARSE) as follows:

NOPROM—Change Promote processing

The NOPROM service modifies the accounting status of a member's accounting record to change the processing of a member when it is copied to the next level during promotion.

For more information on this option, see "New chapter ("Leaving a member behind on promotion")" on page 54.

Command invocation format: The Format of the FLMCMD NOPROM service is:

```
FLMCMD    NOPROM,  
          project,  
          prj_def,  
          group,  
          type,  
          member,  
          accesskey,  
          [REBUILD|REMOVE|NOREBUILD],  
          dd_msgs
```

Call invocation format: The Format of the FLMLNK NOPROM service is:

```

lastrc := FLMLNK('NOPROM',
                sclm_id,
                group,
                type,
                member,
                access_key,
                [REBUILD|REMOVE|NOREBUILD],
                msg_line);

```

Parameters:

- | | |
|-------------------|---|
| project | The project name. The maximum parameter length is 8 characters. |
| prj_def | The project definition name. It defaults to the project name. The maximum parameter length is 8 characters. |
| sclm_id | An SCLM ID associated with a given project and project definition. The INIT service generates the SCLM ID. The maximum parameter length is 8 characters. |
| group | The group at which the member accounting status is to be changed. The maximum parameter length is 8 characters. |
| type | The type of member whose accounting status is to be changed. The maximum parameter length is 8 characters. |
| member | Specifies the member whose accounting status is to be changed. |
| access_key | The access key to be assigned to the member. The access key is required for any further manipulation of the member until you use the UNLOCK service to remove the access key. It defaults to blank. The maximum parameter length is 16 characters. |
| NOREBUILD | By specifying this option, SCLM updates the accounting record to specify an accounting status of NOPROM-N. After building the appropriate member, SCLM while promoting copies the build maps containing the non-promoted member but the member itself is left behind.

If the build map references the member not being promoted, this option causes the build map to be copied during the promote, regardless of the FLMLRBLD macro. |
| REBUILD | By specifying this option, SCLM updates the accounting record to specify an accounting status of NOPROM-R. After building the appropriate member, SCLM while promoting does not copy the build maps containing the non-promoted member or the member itself. Once the copy phase of the promote is complete, the build function is invoked at the level being promoted into. Because the build maps are missing for members that reference the member not being promoted, those members, and any dependent members, are rebuilt.

This option allows you to rebuild at the next level using the editable member found at that level or above it in the hierarchy. |
| REMOVE | By specifying this option, SCLM updates the accounting record to specify an accounting status of EDITABLE. SCLM performs a normal promote of the member. |
| dd_msgs | The ddname indicating the destination of the messages generated by the NOPROM service. If you specify a blank ddname, SCLM routes the NOPROM messages to the default output device, such as your terminal. Otherwise, before you call the NOPROM service, |

you must allocate the ddname. Use the following attributes: RECFM=F, LRECL=80, BLKSIZE=80. The maximum parameter length is 8 characters. This parameter is used for FLMCMD only.

msg_line Services that only write one message have a msg_line parameter. Define a program variable that is 80 characters to hold the contents of this message line. This parameter only applies to services called through the FLMLNK interface.

Chapter 18, "SCLM Macros"

FLMCNTRL macro

- Replace the existing description for the CONTROL parameter with the following:

"Specifies the VSAM data set where the SCLM administrator information and control information is stored. If MEMLOCK=Y, or a value for NOPROMBK is specified, this parameter must be specified."

- Add a new parameter, NPROMBK:
 - In the **Macro format** section, after the CONTROL details, insert:
[,NPROMBK=not_promoted_backup_data_set]
 - In the **Parameters** section, after the description for the CONTROL parameter, insert:

,NPROMBK=not_promoted_backup_data_set

The name of the partitioned data set to contain a backup of unpromoted members. This parameter is optional. If you do not specify a NPROMBK value, unpromoted members are not backed up. No SCLM variables can be used for this parameter.

The FLMNPROM macro must be used to specify which SCLM editable elements can be marked as non-promotable using the NOPROM service.

Note: If the FLMNPROM macro is coded the CONTROL parameter on FLMCNTRL must be specified.

New macro (FLMNPROM)

Add a new section, "FLMNPROM macro" (between the sections for FLMLRBLD and FLMSYSLB) as follows:

FLMNPROM macro: Use this macro to specify which SCLM editable elements may or may not be marked as as non-promotable using the NOPROM service. This macro is required if you want to use the NOPROM service, and can be specified multiple times. Only members that match the groups, types, and languages associated with the NPROM=YES parameter can be marked as non-promotable using the NOPROM service.

Macro format:

```
FLMNPROM
GROUP=(group1,group2,...)|*
,TYPE=(type1,type2,...)|*
,LANG=(lang1,lang2,...)|*
,NPROM=YES|NO
```

Parameters: If NPROM=YES is specified, then GROUPS, TYPES, and LANGUAGES can be used to specify which members can be marked as

NON-PROMOTABLE. If NPROM=NO is specified, then the GROUPS, TYPES, and LANGUAGES can be used to specify which members cannot be marked as NON-PROMOTABLE.

GROUP=(group1,group2,...) | *

The names of the Groups that will be used to specify the members (in combination with the LANG, TYPE, and NPROM parameters) that can be marked as non-promotable. Use an asterisk (*) to indicate all groups.

,TYPE=(type1,type2,...) | *

The names of the Types that will be used to specify the members (in combination with the GROUP, LANG, and NPROM parameters) that can be marked as non-promotable. Use an asterisk (*) to indicate all groups.

,LANG=(lang1,lang2,...) | *

The names of the Languages that will be used to specify the members (in combination with the GROUP, TYPE, and NPROM parameters) that can be marked as non-promotable. Use an asterisk (*) to indicate all groups.

NPROM=YES | NO

Specifies whether any members of which are in any of the Group, Type, and Language combinations, are able to be marked as non-promotable. If NPROM=NO is specified any members that fit the criteria cannot be marked as non-promotable.

Examples: The following example shows how to specify that editable members with a language of COBCOPY in the SOURCE type, in any group, can be marked as non-promotable.

```
FLMNPROM GROUP=*,TYPE=SOURCE,LANG=COBCOPY,NPROM=YES
```

The following example shows how to specify that all editable members with a language of COBCOPY in the SOURCE type, in all groups except EMERFIX can be marked as non-promotable.

```
FLMNPROM GROUP=*,TYPE=SOURCE,LANG=COBCOPY,NPROM=YES  
FLMNPROM GROUP=EMERFIX,TYPE=SOURCE,LANG=COBCOPY,NPROM=NO
```




Printed in USA

sc1m-R678-00



Spine information:



Interactive System Productivity
Facility (ISPF)

Addendum to the SCLM documentation