

SCLM使用の手引き

(SCLM User's Guide)

SCLM使用の手引き

目次

第1部 基本編

第1章 概要	1
1.1 SCLMの構成要素	3
1.2 SCLMの主要な機能	5
1.3 SCLMの3つの定義	6
1.4 グループとタイプ	7
第2章 SCLMの各機能	8
2.1 SCLMメイン・メニュー	9
2.2 編集 (オプション2)	10
2.3 組立て (オプション4)	15
2.4 プロモート (オプション5)	20
2.5 PLMSによるRELLIBへのプロモート及びPL変更	24
2.6 ライブラリー・ユーティリティ (オプション3.1)	36
2.7 移行ユーティリティ (オプション3.3)	38
2.8 アーキテクチャー報告 (オプション3.5)	39
2.9 監査及びバージョン (オプション3.8)	43
第3章 アーキテクチャー定義	45
3.1 アーキテクチャー定義	46
3.2 CC (コンパイル制御) アーキテクチャー・メンバー	47
3.3 LEC (関係編集制御) アーキテクチャー・メンバー	49
3.4 HL (ハイ・レベル) アーキテクチャー・メンバー	51
3.5 Generic (汎用) アーキテクチャー・メンバー	53
3.6 アーキテクチャー定義コーディング・ガイドライン	54
3.7 アーキテクチャー定義 まとめ	55
第4章 プロジェクト定義	56
4.1 プロジェクト定義	57
4.2 代替プロジェクト	62
第5章 言語定義	64
5.1 言語定義	65

SCLM使用の手引き

目次

第2部 応用編

第1章 . バッチ・ロードモジュール編 71

(TSO/IMSネイティブ/CICSネイティブも同様)

1.1 NON-DB2 W/O NCAL	72
1.2 NON-DB2 with NCAL	74
1.3 DB2 W/O NCAL	78
1.4 DB2 with NCAL	80

第2章 . PSB/DBD編 84

2.1 PSB	85
2.2 DBD	87

第3章 . SDTM編 88

3.1 概要	89
3.2 NCALモジュール	95
3.3 O-LOADモジュール	97
3.4 P-LOADモジュール	99
3.5 ROOT-LOADモジュール	102

第4章 . アプリケーション 103

4.1 概要	104
4.2 ロードモジュール	107
4.3 PSB/DBD	108

* APPENDIX *

付1. アーキテクチャー定義作成ツール	110
付2. AMSサービスで使用するSCLMライブラリーのタイプ	113
付3. AMSサービスで使用する言語定義	114
付4. SCLM環境作成ツール一覧	119

I Preface

1. 目的

AMSサービスにおける使用方法を示し、SCLMのマニュアルを補完する。

2. 対象部門

AMSサービス

3. 関連標準

- 21.5.3 SCM計画ガイドライン
- 90.2.3 Production変更手続き
- 70.4.3.1 SCLM管理手続き

4. 区分

リファレンス

5. 発行者

タイトル	担当部門
主管部門	AMSサービス・AMS技術
制定者	AMSサービス担当

6. 文書履歴

改定番号	変更サマリー	日付
Initial		2000/06/30
1.1	修正（1部 3.4、2部 3.5、付3） LE用言語定義の説明追加	2001/10/15

1部 基本編

- 1．概要
- 2．SCLMの各機能
- 3．アーキテクチャー定義
- 4．プロジェクト定義
- 5．言語定義

第1章 概要

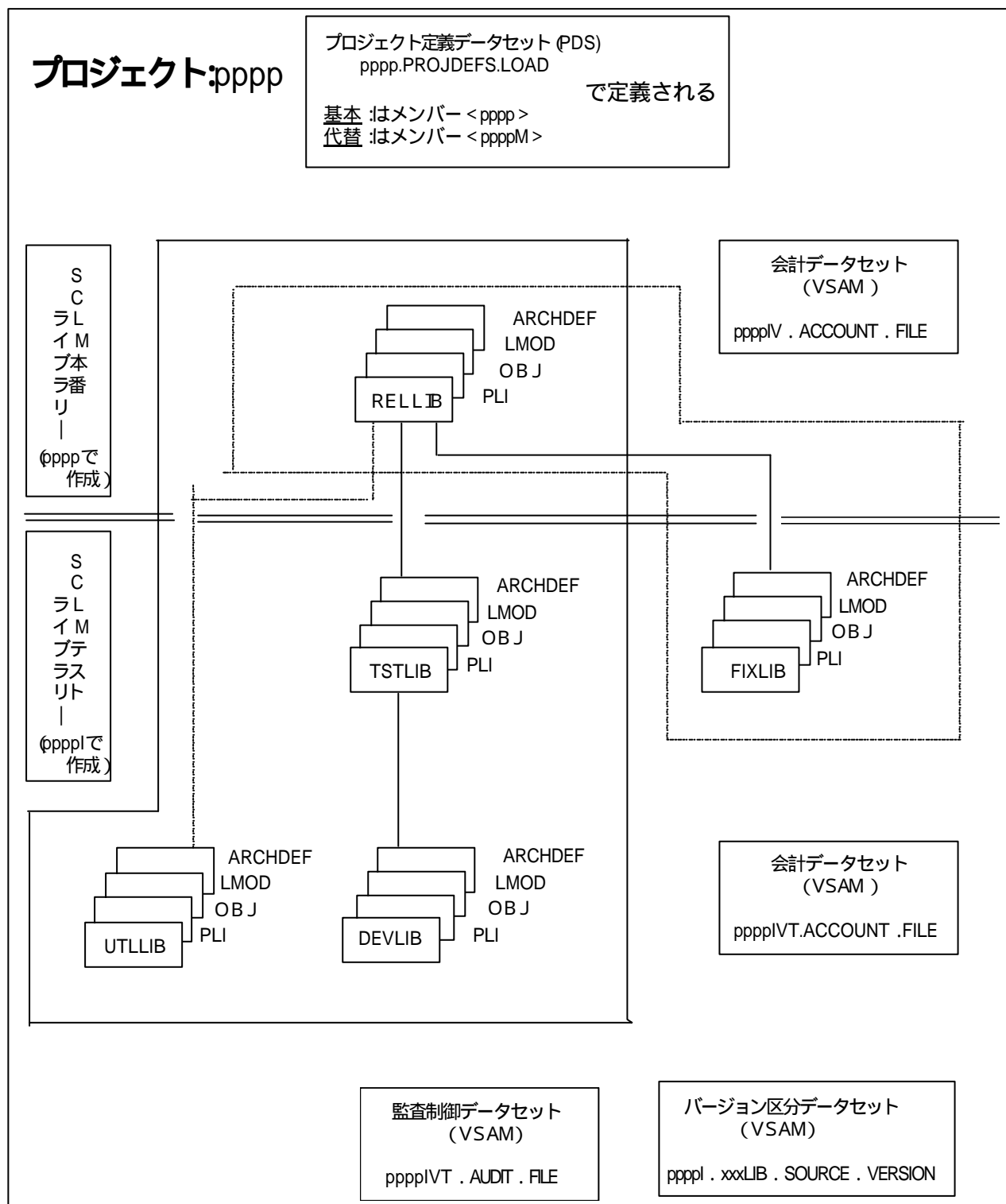
1.1 SCLMの構成要素

1.2 SCLMの主要な機能

1.3 SCLMの3つの定義

1.4 グループとタイプ

1.1 SCLMの構成要素



1.1 SCLMの構成要素

1. SCLMライブラリー (PDS)

本番 : pppp. RELLIB. PLI
テスト : ppppl. TSTLIB. PLI のような形式をとり、それぞれ
 | | |
 プロジェクト グループ タイプ と呼ばれる。
本番ライブラリー (RELLIB) のアロケーションはサービス管理に依頼する。
(「SCLM/PLMS本番ライブラリー登録申請書」を提出)

2. プロジェクト定義データセット (PDS ロードモジュール、プロジェクトに1つ)

プロジェクトに関する定義情報を保管。メンバーによって複数のプロジェクトを定義可能。
pppp.PROJDEFS.LOAD (pppp --- 基本)
 (ppppM --- 代替)

3. 会計データセット (VSAM)

SCLM制御下のメンバーに関する情報を保管する。
本番 : ppppIV.ACCOUNT.FILE
テスト : ppppIVT.ACCOUNT.FILE

4. 監査制御データセット (VSAM)

SCLMの監査機能を使用する場合にのみ必要 (任意指定)
SCLM制御下にある編集可能なメンバーの監査情報が記憶される。
ppppIVT.AUDIT.FILE

5. バージョン区分データセット (PDS)

SCLMのバージョン機能を使用する場合にのみ必要 (任意指定)
SCLM制御下にある編集可能なメンバーのコピーまたはバージョンを保守する。
ppppI.xxxLIB.SOURCE.VERSION

1.2 SCLMの主要な機能

編集（EDIT）機能

SCLM編集機能は、指定されたプロジェクトの階層構造を分析し、その連結順に従い、編集したいメンバーを開発用ライブラリーへコピー（ダウンロード）する。（ここでダウンロードされるのは常に連結の中に存在する最新のバージョンになる）。また、編集セッションが開始されるとSCLMは自動的にそのメンバーをロックする。そして、編集を終了しメンバーをSAVEするとSCLMは変更（編集）され、メンバーとその会計情報を解析し（PARSE）保存（STORE）する。

組立て（BUILD）機能

組立て機能では、まず対象となるアーキテクチャーに定義されたすべての構成要素が存在し、かつ完全であることの検査・確認が行われる。その後、指定された『言語定義』に従い翻訳処理（コンパイル／関係編集）を実行する。

但し、処理される構成要素は、前回の組立て以降に変更のあったメンバーのみである。また、ハイ・レベルのアーキテクチャー定義（「第3章 アーキテクチャー定義」参照で組立てることによって、変更のあったメンバーに依存しているメンバー（変更の必要なメンバー）も自動的に処理される。

プロモート（PROMOTE）機能

プロモート機能は、組立てが行われた構成要素をプロジェクトの階層内の上位グループへと移動させる。また、何らかの関連性のある複数の構成要素については、それらを同時に一度でプロモートさせることも可能である。

監査バージョン機能

この機能はユーティリティーとして提供されている。これを使用すると、SCLM制御メンバーに関する変更の情報やバージョンが保守され、必要に応じそれらを取出すことができる。

1.3 SCLMの3つの定義

アーキテクチャー定義

(開発者が作成)

(アーキテクチャー言語でコーディング)

組立て (BUILD) を実施するための 構成要素や必要な情報 (組立て時の入力・出力・オプション etc..) を定義する。

アーキテクチャー定義は、組立てやプロモートを行う際、その『対象』として指定される。

また、アーキテクチャー定義をネストさせることで『適用業務』の定義が可能となる。

(アーキテクチャー定義は、SCLMライブラリーの1タイプである ARCHDEF に作成する。詳しくは「第3章 アーキテクチャー定義」を参照)

プロジェクト定義

(管理者が作成)

(プロジェクト定義は言語定義と共にアセンブル/リンクされる)

プロジェクト定義はSCLMが制御する開発環境を定義する為のものである。そこでは、SCLMライブラリーの階層構造、プロジェクトの識別名、使用しているライブラリーのグループ・タイプ等が定義される。

その他、プロジェクト固有の制御オプション (カスタマイズ情報) についてもここで定義する。

(プロジェクト定義は、ppppl.PROJDEFS.SOURCE に作成する。詳しくは「第4章 プロジェクト定義」を参照)

言語定義

(管理者が作成)

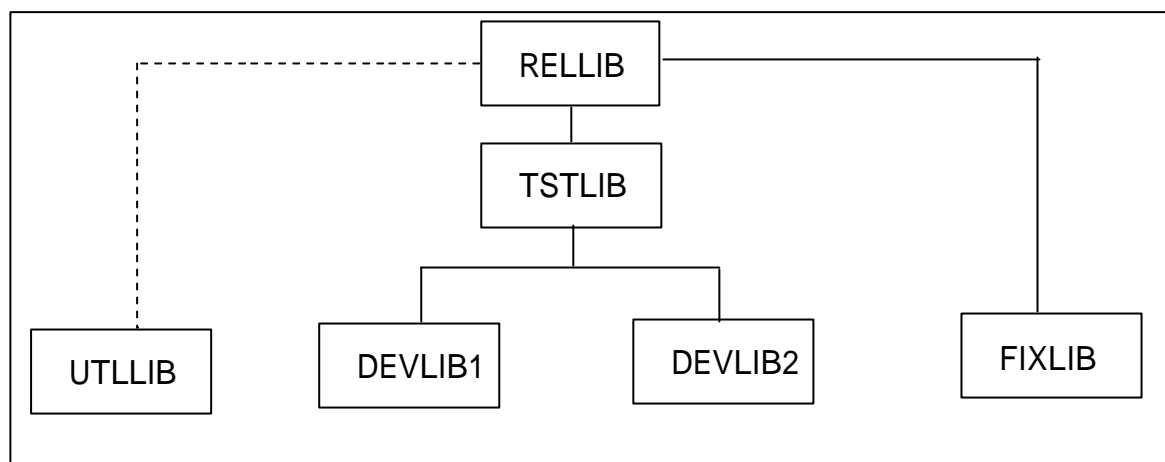
(プロジェクト定義の一部)

通常のコンパイル/リンクにおける JCLのカタプロ に相当し、コンパイラーの名前、DD名、DS名など 翻訳処理を定義 する。また、解析処理 (PARSE) に使用される PARSER もここで定義する。言語定義は 異なる翻訳手順ごとに作成 され、定義した内容はプロジェクト定義と一緒にアセンブル/リンクされる。こうして定義された言語はSCLM制御下の各ソースメンバーに関連付けられ、組立て (BUILD) を実施することによりその翻訳手順が実行される。

(=> コンパイル)

(言語定義は、ppppl.PROJDEFS.SOURCE に作成する。詳しくは「第5章 言語定義」を参照)

1.4 グループとタイプ



1. グループ：同じ第2修飾子を持つプロジェクト・データセットの集合

(例) RELLIB	完成した（移管された）適用業務
TSTLIB	統合テスト用
DEVLIB	各開発者用（AMSサービスでは1つにまとめている）
FIXLIB	トラブル・シューティング用（代替プロジェクト定義ppppMを使用）
UTLLIB	テスト用（ダウンロードのみ可能、 <u>プロモート不可</u> ）

2. タイプ：データの内容を識別するプロジェクト・データセットの第3修飾子

- (1) 異なるレコード形式(RECFM)、レコード長(LRECL)ごとに異なるタイプが必要。
- (2) 例えばCOBOLソースとPLIソースを同じタイプ " PLI " に入れることは可能だが、異なる言語定義を使用する場合、基本としてタイプを分けた方が良い。
- (3) 同じメンバー名を持つ可能性がある場合には、異なるタイプが必要。

基本となる4つのタイプ（詳しくは「* APPENDIX * 付2」参照）

(例) PLI	PLIソース・コード用
OBJ	オブジェクト・コード用（オブジェクトを保管しない方法もある）
LMOD	ロード・モジュール用
ARCHDEF	アーキテクチャー用

第2章 SCLMの各機能

- 2.1 SCLM基本オプション・メニュー
- 2.2 編集（オプション2）
- 2.3 組立て（オプション4）
- 2.4 プロモート（オプション5）
- 2.5 ライブラリー・ユーティリティー（オプション3.1）
- 2.6 PLMSによるRELLIBへのプロモート及びPL変更
- 2.7 移行ユーティリティー（オプション3.3）
- 2.8・アーキテクチャー報告（オプション3.5）
- 2.9 監査及びバージョン（オプション3.8）

2.1 SCLMメイン・メニュー

SCLM メイン・メニュー	
オプション ===>	
以下のオプションのいずれかを入力してください。	
1 ビュー	データの ISPF ビューまたは表示
2 編集	SCLM データベースのソースを作成または変更してください
3 ユーティリティー	SCLM データベース・ユーティリティーまたは報告書機能の実行
4 作成	SCLM に制御された構成要素の構成
5 プロモート	SCLM 階層へ構成要素を移動
6 コマンド	TSO または SCLM コマンドを入力
X 終了	SCLM の終了
SCLM プロジェクト制御情報 :	
プロジェクト . . . TD77	(プロジェクト高水準修飾子)
代替	(プロジェクト定義 : プロジェクトの省略時値)
グループ DEVLIB	(TSO 接頭部の省略時値)

1. オプションの説明

- | | |
|---------------------------|--------------------------------------|
| (1) ビュー - (BROWSE) | ISPF/PDFの表示 |
| (2) 編集 (EDIT) | ISPF/PDFの編集機能
SCLM管理下のメンバーの編集には必須 |
| (3) ユーティリティー | ライブラリーの保守、会計情報管理、報告書 |
| (4) 作成 | モジュールのコンパイル/リンク |
| (5) プロモート | メンバーの上位階層への移動 |
| (6) コマンド | TSOまたはSCLMコマンドを入力 |

2. 入力フィールドの説明

プロジェクト: プロジェクト名 = データセットの第1修飾子 (例: TD77)

グループ: TSO接頭部の省略時値 (例: DEVLIB)

2.2 編集(オプション2)

SCLM 編集 - 入力パネル		続く	+
SCLM ライブラリー :			
プロジェクト	: TD77		
グループ	: DEVLIB TSTLIB RELLIB		
タイプ	: PLI		
メンバー	: (メンバー選択リストは、ブランクかパターン)		
初期マクロ . . .			
プロファイル名	: (ブランクの場合、データ・セット・タイプに省略時値)		
オプション			
/	取消し / 移動 / 置換を確認		
/	混合モード		
	ワークステーションでの編集		
変更コード	:		
許可コード	: (ブランクの場合、省略時値許可コードが使用されます)		
パーサー・ボリューム	: (ブランクの場合、省略時値ボリュームが使用されます)		
コマンド	====>		

1. 主な入力フィールドの説明

グループ : 4つあるうちの最初のフィールドはSCLM基本メイン・メニューで指定した開発グループになり変更できない。SCLMは残りの3つのフィールドに(もしあれば)階層中の次のグループを自動的に表示する。
この3フィールドについては変更可能である。しかし正しくないグループを指定したり、連結順序を誤って指定したりするとエラーとなる。

変更コード : 未使用のためブランク

許可コード : 省略時値を使用のためブランク

パーサー・ボリューム : 指定不要のためブランク

2.2 編集(オプション2)

2. メンバー・リスト

メンバー名にブランクまたはパターンを使用すると下記のメンバー・リストが表示される。リストの“LIB”フィールドには連結順序の中のどのライブラリーにメンバーが存在するかが表示される。

EDIT	TD77.DEVL					LIB1.PLI				行 00001 / 00004			
コマンド		====>					スクロール ===> PAGE						
名前		LIB	VV	MM	作成日		最終変更日		SIZE	INIT	変更	ID	
. PLIPGM		2	01.02	93/11/09	97/05/23	11:51	5	3	2	EXXXXX			
. TEST1		2	01.00	00/05/22	00/05/22	20:59	4	4	0	EXXXXX			
. TEST2		2	01.01	00/05/22	00/05/22	21:00	4	4	2	EXXXXX			
. TEST3		2	01.01	00/05/22	00/05/22	21:00	4	4	2	EXXXXX			

3. 言語の割り当て

新規メンバーを編集した場合、SAVEする際には必ず下記の画面が表示される。ここで“言語”フィールドにあらかじめ定義してある「言語定義名」を指定する。既存のメンバーの「言語」を変更する場合にはEDIT中にSPROFコマンドを入力すると下記の画面が表示される。

メニュー	SCLM	ユーティリティー	ヘルプ

メンバー : ATEST			
希望する値を入力し、ENTER を押してください。			
変更しないで終了するには、CANCEL コマンドを入力してください。			
言語 PLI2		
変更コード	. . . (最新入力呼び出すには "=" を使用してください)		

2.2 編集(オプション2) ...

4. 「言語」とは...

SCLM制御下のメンバーにはすべて、「言語」が割り当てられる必要がある。「言語」はそのメンバーが組立て時にどのコンパイラーによって処理されるかを決定するもので、プロジェクト毎に事前に言語定義により定義しておく。

例えば、同じPLIのソースでもSQL文を含むものはPLIコンパイラーで処理する前にDB2のプリコンパイラーでの処理が必要なため、SCLMでは異なる言語定義を使用する。つまり「言語」は異なる処理(コンパイル)手順を必要とするメンバーごとに定義され、割り当てられる。

「言語」は各メンバーごとに割り当てられるもので、そのメンバーが保管されているライブラリーのタイプ(例:PLI)とは無関係である。例えば、PLIというタイプのライブラリーの中にCOBOLコンパイル用の言語定義名を割り当ててあれば問題なく組立てが行われる。(但し基本的には、ソースの種類～言語の種類～によってタイプを分けて使用する。)

「言語」の割り当ては編集可能なメンバー(ソース)に対してのみ行われ、OBJなどの組立ての結果得られるものについては、ソースの言語が引き継がれる。(但し、ロード・モジュールは例外で独自の言語が割り当てられる。)

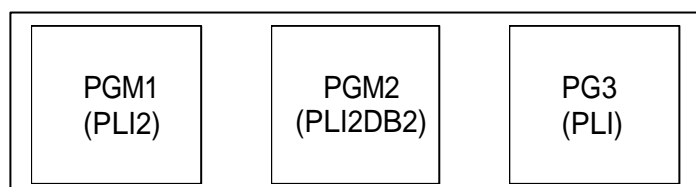
編集可能メンバーに「言語」を割り当てるには、以下の3つの方法がある。

1. 新規作成メンバーのSAVE時にプロンプトに答える。
2. 既存メンバーに対して編集中にSPROFコマンドを使用して変更する
3. SCLM外で作成されたメンバーに移行ユーティリティー(3.3)を使用する
(3は初回のみ有効)

各メンバーに割り当てられた言語名は会計情報として保管され、ライブラリー・ユーティリティー(オプション3.1)で確認できる。

(例) PROJ01.DEVLIB1 . PLI

言語の例



PLI2

通常のPL/ I Compiler(Ver . 2.2)

PLI2DB2

DB2 PreCompile + PLI2

PLIC

CICS Translate + PLI2

(詳しくは、「* APPENDIX * 付3」を参照)

2.2 編集(オプション2) ...

5. SCLM編集機能とISPF/PDF編集機能との違い

SCLM制御下のメンバーの編集には必ずSCLM編集機能を使用する。

ISPF/PDFの編集を使用すると警告メッセージが出される。

==> 編集中メンバーはSCLMに制御されているプロジェクトに属する。

さらにメンバーがすでに会計情報をもっている場合は以下の=>MSGが表示される。

==> MSG > -注意-このメンバーを保管するとSCLMの会計情報が無効になる。

6. PDF編集機能との相違点

(1) メンバーのロック :

SCLMは編集セッションを開始するとメンバーをロックし、同じプロジェクト内の他の開発ライブラリーでの編集を防ぐ。

(例 : DEVLIB1で編集するとDEVLIB2では編集不可)

(2) SAVEコマンド :

メンバーを保管する前に、メンバーを解析 (PARSE) し、結果を会計データセットに保管 (STORE) する。

解析機能はISPF/PDFのパック形式を認識しない。(PACK OFFにしておく)

(3) SCREATE/SMOVE/SREPLACE コマンド :

通常の編集におけるCREATE, MOVE, REPLACEの機能に加えてメンバーのロック、解析、会計情報の作成を行う。これらのコマンドには必ず一緒にメンバー名を指定する。(メンバー名を省略してもメンバー指定のためのパネルは表示されないので注意。)

(4) SPROFコマンド :

メンバーに割り当てる「言語」を指定/変更する。

編集可 / 不可の認識

SCLMは組立ての出力として得られるタイプ(OBJ, LOAD, LIST ...)は、たとえそれが実際に編集可能であっても、編集不可として扱う。

2.2 編集(オプション2) ...

7. もしもSCLM編集機能以外で変更した場合は

会計情報が正しく更新されていないのでSCLM編集機能でオープンしSAVEコマンドで強制的にSAVEすることで会計情報を更新する。これを実施しないと組立て時にエラーとなる。

(FLM07007 : ACCOUNT INFORMATION IS NOT ACCURATE)

8. SCLMライブラリー拡張での注意点

SCLM制御下で編集されたメンバーを新しく拡張したSCLMライブラリー上にCOPYする際、ISPF/PDFの移動・複写ユーティリティ(オプション3.3)を使用するとSCLMの会計情報が無効になるので注意する。

会計情報を保持する為には「IEBCOPYユーティリティ」を使用すること。

2.3 組立て/BUILD (オプション4)

SCLM 組立て - 入力パネル			
コマンド ==>			
組立て入力 :			
プロジェクト	:	TD77	
グループ	:	DEVLIB1	
タイプ	:	ARCHDEF	選択には、"/" を入力
メンバー	:	HL@TEST	/ エラー・リストのみ ワークステーション作成
モード	1	1. 条件付き 2. 無条件 3. 強制 4. 報告書	有効範囲 2 1. 制限 2. 通常 3. サブユニット 4. 拡張
出力制御 :			
	EX SUB		処理 . . . 1 1. 実行 (EX) 2. サブミット (SUB)
メッセージ	3 3	1. 端末	
報告書	. . 3 3	2. 印刷装置	
リスト	. . 3 3	3. データセット	印刷装置 . . . H
		4. なし	ボリューム . .

1. 主な入力フィールドの説明

タイプ : 通常はARCHDEFを指定する。それ以外にはソースの指定が可能。

有効範囲 : 常に ' 2. 通常 ' を指定。

組立てモード : 条件付き 指定された以上の戻りコード (言語定義のGOODRC)
処理を停止 (省略時値)。

無条件 戻りコードの値にかかわらず実行
複数の構成要素でのエラーを検出したい場合に使用。

強制 要求された構成要素すべての再組立てを実行。
特別な場合以外は使用しない。

報告書 実際に組立てを実行させずに、無条件モードを
シミュレートし報告書を出力する。

2.3 組立て/BUILD (オプション4)

2. 組立てとは

組立てとは、コンパイル/リンクを自動的に実行する機能である。作成、BUILDとも言われる。組立て処理は下記のような手順で行われる。

3. 組立て処理の各段階

(1) 検査 (Verification)

組立ての対象となるすべてのソースおよびアーキテクチャー定義が、正しい会計情報を持っているかを検査する。

(2) 依存関係処理 (Dependency Processing)

) 組立てマップの作成

組立ての対象となる構成要素に対して組立てマップを作成する。組立てマップにはその構成要素を作成するのに参照されたすべての構成要素の名前と、それらの最終更新日付が含まれる。

) 翻訳機能 (コンパイラ) の呼び出し

組立てマップの情報と会計情報をもとに、最後の組立て以降に変更が行われたかを判定し、変更されたもの、およびそれによって影響されるものについて翻訳機能 (コンパイラ) を呼び出す。

) 報告書作成

組立ての結果を報告書として出力する。

SCLMは変更が必要なものだけに対して、最小限のコンパイル/リンクのみ実行する

(参考)SCLMがモジュール間の依存関係を判断する方法

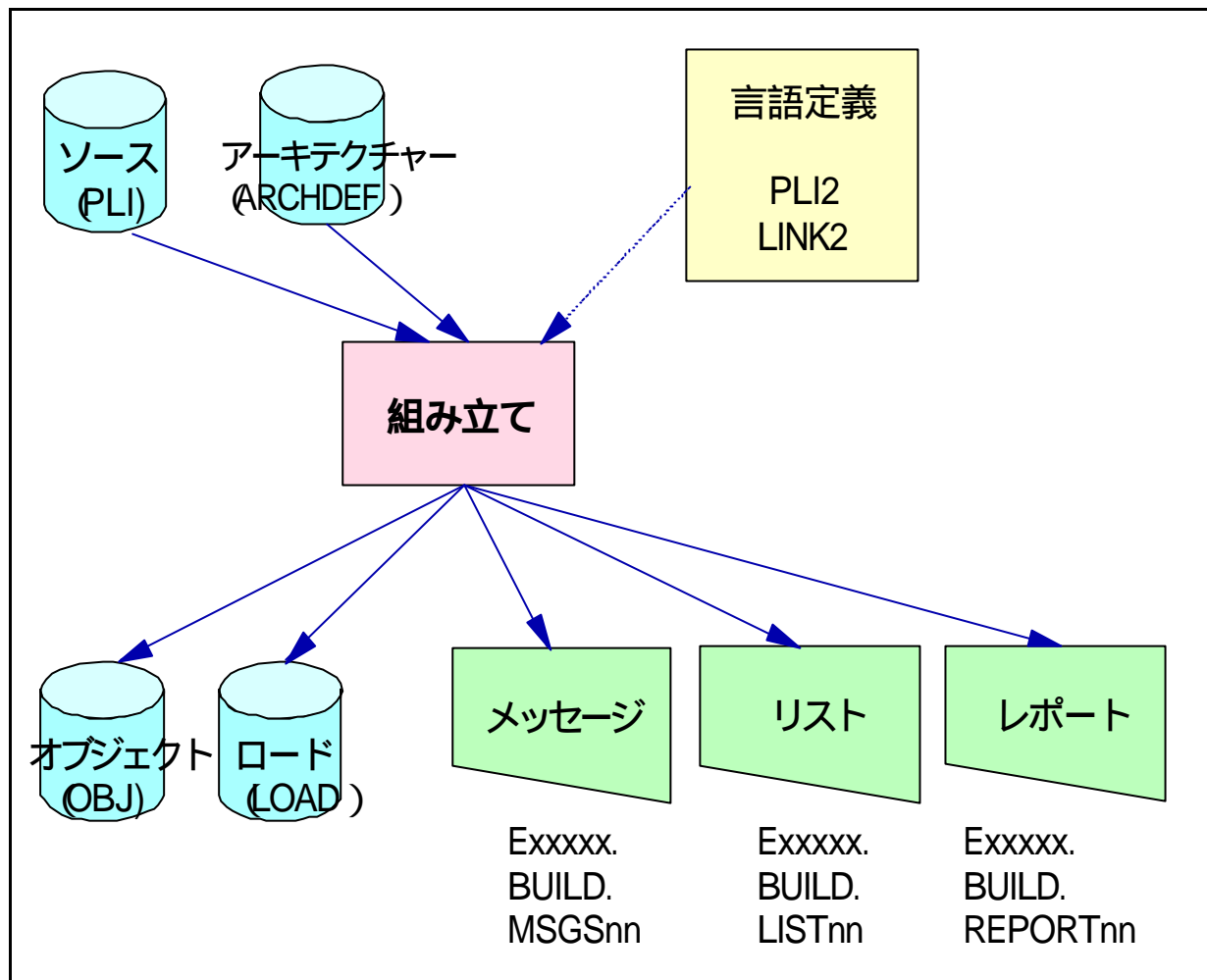
1) 組み込みメンバーにおける依存関係

PLIの%INCLUDE、COBOLのCOPYなどの組み込みメンバーに関する依存関係はメンバーの解析時にPARSERによって認識され、会計データセットに保管される。

2) 外部参照における依存関係

外部ルーチンのCALLを行っている場合は、アーキテクチャー定義で外部ルーチンを一緒に連係編集するように指定することで、依存関係が認識される。

2.3 組立て/BUILD (オプション4)



組立て報告書 出力例

```

*****
*****
**
**
**      SOFTWARE CONFIGURATION AND LIBRARY MANAGER (SCLM)
**
**      B U I L D   R E P O R T
**
**
**      00/05/24   17:31:19
**
**
**      PROJECT:      TD77
**      GROUP:        DEVLIB
**      TYPE:         ARCHDEF
**      MEMBER:       HL@TEST
**      ALTERNATE:    TD77
**      SCOPE:        NORMAL
**      MODE:         CONDITIONAL
**
**
*****
*****
1   ***** B U I L D   O U T P U T S   G E N E R A T E D ***** PAGE 1

MEMBER      TYPE      VERSION      KEYWORD
-----
TEST1       OBJ        1          OBJ
TEST2       OBJ        1
TEST3       OBJ        1
TEST1       LMOD        1          LOAD
TEST2       LMOD        1
TEST3       LMOD        1
1   ***** B U I L D   M A P S   G E N E R A T E D ***** PAGE 2

                                     (REASON FOR REBUILD)
MEMBER      TYPE      VERSION      MEMBER      TYPE
-----
HL@TEST     ARCHDEF    1          *** NEW BUILD MAP GENERATED

```

TEST1	ARCHDEF	1	*** NEW BUILD MAP GENERATED
TEST2	ARCHDEF	1	*** NEW BUILD MAP GENERATED
TEST3	ARCHDEF	1	*** NEW BUILD MAP GENERATED
TEST1	PLI	1	*** NEW BUILD MAP GENERATED
TEST2	PLI	1	*** NEW BUILD MAP GENERATED
TEST3	PLI	1	*** NEW BUILD MAP GENERATED

2.4 プロモート(オプション5)

SCLM プロモート - 入力パネル			
プロモート入力 :			
プロジェクト : TD77			
開始グループ DEVLIB			
タイプ ARCHDEF			
メンバー ATEST			
モード 1		1. 条件付き	有効範囲 1 1. 通常
		2. 無条件	2. サブユニット
		3. 報告書	3. 拡張
出力制御 :			
EX SUB		処理 . . . 1	1. 実行 (EX)
メッセージ 3 3	1. 端末	2. サブミット (SUB)	
報告書 . . 3 3	2. 印刷装置	印刷装置 . . . H	
	3. データセット	ボリューム . .	
	4. なし		

1. 主な入力フィールドの説明

タイプ : 通常ARCHDEFを指定する。アーキテクチャー定義によって参照されるすべての構成要素が一括してプロモートされる。

有効範囲 : 常に ' 1. 通常 ' を指定。

プロモートモード : 条件付き 検査エラーがあるCOPY, PURGEを行わない。(省略時)

無条件 検査エラーがあってもCOPY, PURGEを実行する。
部分的に完成した適用業務をプロモートする場合等に使用する
(使用に当たっては十分注意する)。

報告書 検査、報告書作成のみ実行 (PL変更前には必須)。

2.4 プロモート(オプション5)

2. プロモートとは

メンバーを任意のグループからその次に高いグループに移行する機能。
例えばDEVLIB TSTLIBあるいはTSTLIB RELLIBなどへの移行がこれにあたる。
プロモートは下記のような手順で行われる。

3. プロモートの4つの段階

(1) 検査 (Verification)

アーキテクチャー定義によって参照される(プロモートの対象となる)すべての構成要素に対して、それらが最新で(最後のBUILDから変更されていない)正しい会計情報をもっているかを検査する。

(2) 複写 (COPY)

階層の次のグループにメンバーを複写する。複写はタイプごとにまたタイプのアルファベット順に行われる。途中でエラー(スペース不足など)が発生した場合は、原因を取り除いた後、再度プロモート。完了した作業は繰り返されない。

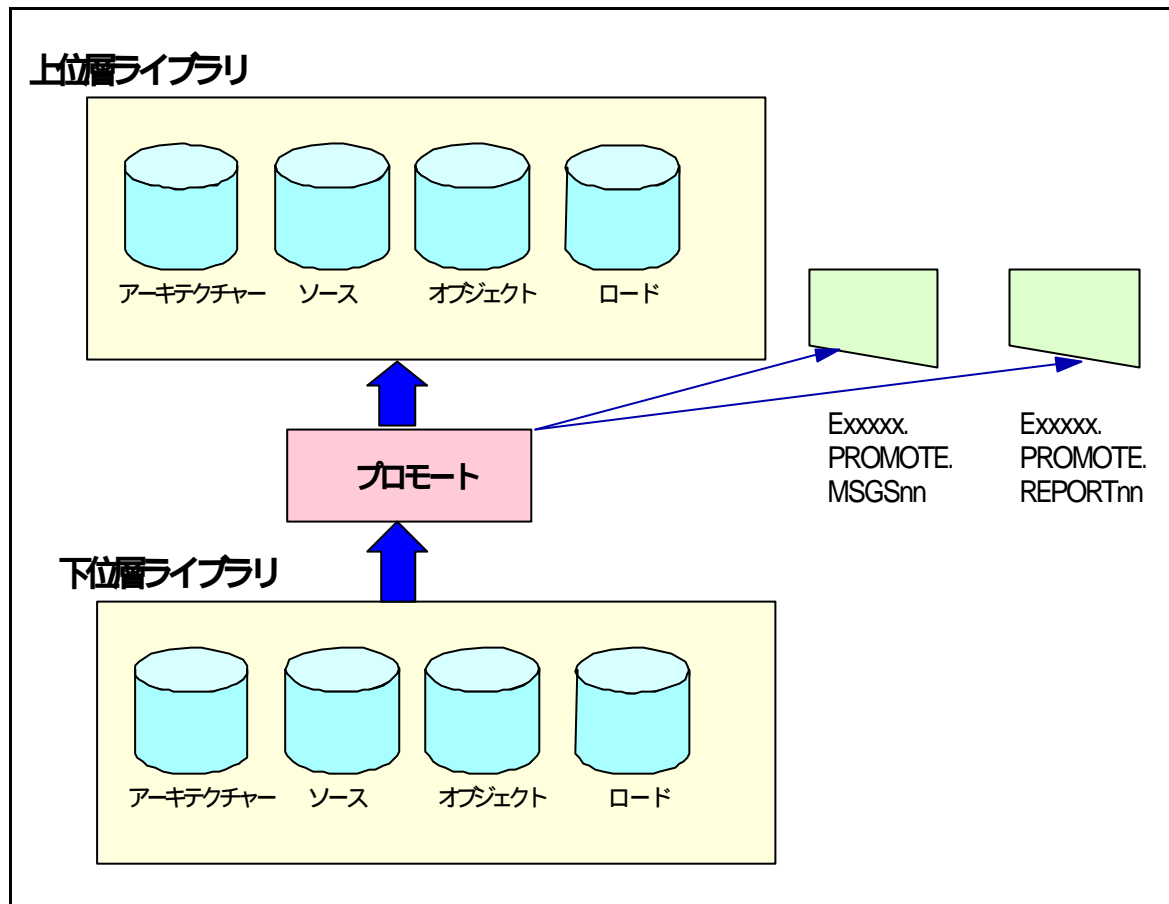
(3) 除去 (Purge)

複写が正常に完了してから、下位のグループからメンバーを削除する。

(4) 報告書作成

検査でエラーとなった場合は、プロモート可能なメンバーのリストが表示され、検査がOKの場合はプロモートの結果が表示される。

2.4 プロモート(オプション5)



プロモート報告書出力例

プロモートされたメンバーが次のグループに複写され、元のグループから削除されたことがX印で示されている。

**					**
**					**
**	SOFTWARE CONFIGURATION AND LIBRARY MANAGER (SCLM)				**
**					**
**	P R O M O T E R E P O R T				**
**					**
**	00/05/24	17:36:33			**
**					**
**					**
**	PROJECT:	TD77			**
**					**
**	TO GROUP:	TSTLIB			**
**	FROM GROUP:	DEVLIB			**
**					**
**	TYPE:	ARCHDEF			**
**	ARCH. MEM.:	HL@TEST			**
**	ALTERNATE:	TD77			**
**	SCOPE:	NORMAL			**
**	MODE:	CONDITIONAL			**
**					**
**					**
**	NOTE: "*" INDICATES "OUT OF SCOPE" ITEMS.				**

1					PAGE 2
TYPE: ARCHDEF					
				COPIED TO	PURGED FROM
MEMBER	DATE	TIME	MESSAGE	TSTLIB	DEVLIB

HL@TEST	00/05/24	17:30:00		X	X
TEST1	00/05/24	17:27:00		X	X
TEST2	00/05/24	17:27:00		X	X
TEST3	00/05/24	17:28:00		X	X
1					PAGE 3
TYPE: LMOD					
				COPIED TO	PURGED FROM

24

TYPE: PLI

MEMBER	DATE	TIME	MESSAGE	COPIED TO TSTLIB	PURGED FROM DEVLIB
TEST1	00/05/24	17:31:19		X	X
TEST2	00/05/24	17:31:19		X	X
TEST3	00/05/24	17:31:19		X	X

2.5 PLMSによるRELLIBへのプロモート及びPL変更

AMSサービスでは、RELLIBへのプロモートおよびPL変更をPLMSで行うようになっている。PL変更の前には、報告書モードでプロモートされるメンバーを確認する。

報告書モードでのプロモート

SCLM プロモート - 入力パネル			
プロモート入力 :			
プロジェクト : TD77			
開始グループ TSTLIB			
タイプ ARCHDEF			
メンバー HL@TEST			
モード <u>3</u>		1. 条件付き	有効範囲 1 1. 通常
		2. 無条件	2. サブユニット
		3. 報告書	3. 拡張
出力制御 :			
EX SUB		処理 . . . 1	1. 実行 (EX)
メッセージ 3 3	1. 端末		2. サブミット (SUB)
報告書 . . 3 3	2. 印刷装置	印刷装置 . . . H	
	3. データセット	ボリューム . .	
	4. なし		

プロモート報告書（報告書モード）

プロモート可能なメンバーの一覧が表示される。

実際にコピーは行われないため、X印は記されない。

```
*****
*****
**
**
**      SOFTWARE CONFIGURATION AND LIBRARY MANAGER (SCLM)
**
**
**      P R O M O T E   R E P O R T
**
**      00/05/24   20:50:10
**
**
**
**      PROJECT:      TD77
**
**
**      TO GROUP:      RELLIB
**      FROM GROUP:    TSTLIB
**
**
**      TYPE:          ARCHDEF
**      ARCH. MEM.:    HL@TEST
**      ALTERNATE:      TD77
**      SCOPE:          NORMAL
**      MODE:           REPORT
**
**
**
** NOTE: "*" INDICATES "OUT OF SCOPE" ITEMS.
*****
*****

1                                     PAGE    2

TYPE: ARCHDEF

MEMBER      DATE      TIME      MESSAGE      COPIED TO      PURGED FROM
             DATE      TIME      MESSAGE      RELLIB        TSTLIB
-----
HL@TEST     00/05/24   17:30:00
TEST1       00/05/24   17:27:00
TEST2       00/05/24   17:27:00
TEST3       00/05/24   17:28:00

1                                     PAGE    3

TYPE: LMOD

COPIED TO      PURGED FROM
```

MEMBER	DATE	TIME	MESSAGE	RELLIB	TSTLIB
TEST1	00/05/24	17:32:00			
TEST2	00/05/24	17:32:00			
TEST3	00/05/24	17:32:00			
1					PAGE 4
TYPE: OBJ					
MEMBER	DATE	TIME	MESSAGE	COPIED TO RELLIB	PURGED FROM TSTLIB
TEST1	00/05/24	17:31:00			
TEST2	00/05/24	17:31:00			
TEST3	00/05/24	17:32:00			
1					PAGE 5
TYPE: PLI					
MEMBER	DATE	TIME	MESSAGE	COPIED TO RELLIB	PURGED FROM TSTLIB
TEST1	00/05/24	17:25:00			
TEST2	00/05/24	17:25:00			
TEST3	00/05/24	17:26:00			
1					PAGE 6

**					
**					
B U I L D M A P S					
**					

1					PAGE 7
TYPE: ARCHDEF					
MEMBER	DATE	TIME	MESSAGE	COPIED TO RELLIB	PURGED FROM TSTLIB
HL@TEST	00/05/24	17:31:19			
TEST1	00/05/24	17:31:19			
TEST2	00/05/24	17:31:19			
TEST3	00/05/24	17:31:19			

1

PAGE 8

TYPE: PLI

MEMBER	DATE	TIME	MESSAGE	COPIED TO RELLIB	PURGED FROM TSTLIB
TEST1	00/05/24	17:31:19			
TEST2	00/05/24	17:31:19			
TEST3	00/05/24	17:31:19			

2.5 PLMSによるRELLIBへのプロモート及びPL変更

1. PLMSでのプロモートの方法

確認が終わったら、PLMSで移管を行う。

PLMSでの操作方法是以下の通り。

```

----- プロダクション・ライブラリー変更管理システム ----- 00/05/24 - 19:03
オプション ==> P

ブランク - 申請のリスト表示 (アプリケーション番号指定)
B - 申請の表示 (申請番号指定) R - PL変更の申請
D - 申請の取消し (申請番号指定) P - SCLMプロモート及
G - Go-Sign の入力 (申請番号指定) PL変更の申請

申請番号 ==> (オプション B,D,G の場合)

アプリケーション番号 ==> (オプションがブランクの場合)

申請内容 < 以下の項目を記入 > (オプション R,P の場合)
移管申請番号 ==>
アプリケーション番号 ==> TD77
PROJECT または PSR 番号 ==> TD771001 R A T S 問題管理番号 ==>
申請理由 ==> ????????
変更開始日 ==> 000524 (YYMMDD)
本番反映日 ==> 000524 (YYMMDD)

タイミング・オプション ==> N (Y:YES / N:NO)
緊急処理 ==> N (Y:YES / N:NO)
  
```

- (1) PLMS初期画面よりオプション 'P' を選択し、通常のPL変更と同様に必要事項を入力して、Enterキーを押す。

2.5 PLMSによるRELLIBのプロモート及びPL変更

----- SCLMプロモート及びP L 変更申請入力 (P L M S) -----

移管申請番号 :
アプリケーション番号 : TD77
PROJECT または PSR 番号 : TD771001 R A T S 問題管理番号 :
申請理由 : ????????
変更実施日 : 000524 本番反映日 : 000524
緊急処理 : N タイミング・オプション : N

< 申請するプロモートの内容 >

プロモートするアーキテクチャー定義のメンバー名 ==> HL@TEST

代替プロジェクト名 ==> TD77

- (2) プロモートするアーキテクチャー定義のメンバー名と代替プロジェクト番号を入力して、Enterキーを押す。

2.5 PLMSによるRELLIBへのプロモート及びPL変更

```

----- SCLMプロモート及びP L 変更申請入力 ( P L M S ) -----

移管申請番号      :
アプリケーション番号 : TD77
PROJECT または PSR 番号 : TD77I001   R A T S 問題管理番号  :
申請理由          : ????????
変更実施日        : 000524      本番反映日          : 000524
緊急処理          : N           タイミング・オプション : N

< 申請するプロモートの内容 >
----- 申請内容の確認 -----
プロモ- | 申請するプロモートの内容に間違いはありませんか？ |
         | 移動元ライブラリー   : TD77.TSTLIB.*             |
         | 移動先ライブラリー   : TD77.RELLIB.*             |
代替プロ | 代替プロジェクト名   : TD77                         |
         | 定義メンバー名      : HL@TEST                       |
         | 下記のオプションを指定して下さい。                   |
         | P : この内容で申請を行う。                       |
         | C : 申請内容を訂正する。                         |
         | Q : この申請をとりやめる。                       |
         | オプション ==>                                     |
コマンド | -----

```

- (3) プロモートの内容に間違いがなければ、'P'を、内容(メンバー名または代替プロジェクト番号)を訂正したい場合は'C'を、また申請を中止する場合は'Q'を入力してEnterキーを押す。

2.5 PLMSによるRELLIBへのプロモート及びPL変更

----- プロダクション・ライブラリー変更管理システム ----- REQ# : 00001900
オプション ==> P

ブランク - 申請のリスト表示 (アプリケーション番号指定)

B - 申請の表示 (申請番号指定) R - PL変更の申請

D - 申請の取消し (申請番号指定) P - SCLMプロモート及

G - Go-Sign の入力 (申請番号指定) PL変更の申請

申請番号 ==> (オプション B,D,G の場合)

アプリケーション番号 ==> (オプションがブランクの場合)

申請内容 < 以下の項目を記入 > (オプション R,P の場合)

移管申請番号 ==>

アプリケーション番号 ==> TD77

PROJECT または PSR 番号 ==> TD771001 R A T S 問題管理番号 ==>

申請理由 ==> ????????

変更開始日 ==> 000524 (YYMMDD)

本番反映日 ==> 000524 (YYMMDD)

タイミング・オプション ==> N (Y:YES / N:NO)

緊急処理 ==> N (Y:YES / N:NO)

(4) プロモート申請が受けられると申請番号が表示されるので、後は通常のPL変更と同様の手続を行う。

プロモートは、タイミング・オプションに'Y'を指定していれば、Goサインを入力した時に、そうでなければコンファームされたときにバッチジョブで行われる。

ジョブ名はExxxnP(Exxxxは申請者のユーザーID、nは申請番号の下一桁)ですので結果を確認すること。

2.5 PLMSによるRELLIBへのプロモート及びPL変更

```

*****データの始め*****
JES2 JOB LOG -- SYSTEM QHEA -- NODE

13.36.43 JOB02528 ---- FRIDAY, 02 JUN 2000 ----
13.36.43 JOB02528 ICH70001I DMSSX LAST ACCESS AT 13:35:31 ON FRIDAY, JUNE 2,
13.36.43 JOB02528 $HASP373 Exxxxx1P STARTED - INIT 76 - CLASS J - SYS QHEA
13.36.43 JOB02528 IEF403I Exxxxx1P - STARTED - TIME=13.36.43
13.36.43 JOB02528 - --TIMINGS (MINS.)--
13.36.43 JOB02528 -JOBNAME STEPNAME PROCSTEP RC EXCP CPU SRB CLOCK
13.36.43 JOB02528 -Exxxxx1P SCRATCH A 00 34 .00 .00 .0
13.36.43 JOB02528 -Exxxxx1P SCRATCH B 00 30 .00 .00 .0
13.36.44 JOB02528 -Exxxxx1P GENSGFI 00 65 .00 .00 .0
13.36.44 JOB02528 -Exxxxx1P BATCH GENERO 00 13 .00 .00 .0
13.37.07 JOB02528 -Exxxxx1P BATCH PROMOTE 00 7800 .03 .00 .3
13.37.20 JOB02528 -Exxxxx1P SCLG 00 8289 .01 .00 .2
13.37.21 JOB02528 -Exxxxx1P LTQ 00 175 .00 .00 .0
13.37.23 JOB02528 -Exxxxx1P SK57Q10 00 71 .00 .00 .0
13.37.26 JOB02528 -Exxxxx1P SK57Q22 00 598 .00 .00 .0
13.37.26 JOB02528 +PLMS001I UID=Exxxxx,PL=P0001821
13.37.26 JOB02528 -Exxxxx1P SK57MSG 00 12 .00 .00 .0
13.37.26 JOB02528 IEF404I Exxxxx1P - ENDED - TIME=13.37.26
13.37.26 JOB02528 -Exxxxx1P ENDED. NAME-@DM@Exxxxx TOTAL CPU TIME=
13.37.26 JOB02528 $HASP395 Exxxxx1P ENDED
----- JES2 JOB STATISTICS -----

```

(5) ジョブの結果確認

プロモートは、タイミング・オプションに 'Y' を指定していればGoサインを入力したときに、そうでなければコンファームされたときにバッチジョブで行われる。

ジョブ名はExxxxxnP (Exxxxxは申請者のユーザーID、nは申請番号の下一桁) なので、結果を確認する。

なお、'SK57.DS.DSLIST(pppp)' が空の場合は、'LTQ' というステップで戻りコード4が立ち、その後のステップは迂回されてジョブは終了する。

PLMSの申請状況は 'CONFIRMED' のままになるが、RELLIBへのプロモートは行われる。

2.5 PLMSによるRELLIBへのプロモート及びPL変更

DSL IST - 一致 DATASET DMSSI.PROMOTE.REPORT		行 1 / 5
コマンド - "/" を入力してアクションを選択	メッセージ	VOLUME

DMSSI.PROMOTE.REPORT.#001630		EASS08
DMSSI.PROMOTE.REPORT.#001651		EASS10
DMSSI.PROMOTE.REPORT.#001670		EASS09
DMSSI.PROMOTE.REPORT.#001672		EASS07
DMSSI.PROMOTE.REPORT.#001676		EASS10
***** データセット・リストの終り *****		

(6) プロモートの結果確認

プロモートの結果はDMSSI.PROMOTE.REPORT.#xxxxxx(xxxxxxは申請番号の下六桁)というデータセットで確認できる。もしもこのデータセットが空の場合はDMSSI.PROMOTE.MESSAGE.#xxxxxxというデータセットでメッセージを確認する。

2.6 ライブラリー・ユーティリティー(オプション3.1)

SCLM管理下の各メンバーについて、会計レコード、組立てマップの表示、削除を行う
メンバー・リストの表示

メンバー・リスト TD77.DEVLIB.PLI - 階層ビュー -				メンバー 中の 298		
A= 会計 M= マップ B= 表示 D= 削除 E= 編集 V= ビュー C= 組立て U= 更新						
M.....	状況	TEXT	CHG DATE	CHG TIME	ACCOUNT	BLD MAP
TEST1		TSTLIB	00/05/24	17:25:00	TSTLIB	TSTLIB
TEST2		TSTLIB	00/05/24	17:25:00	TSTLIB	TSTLIB
TEST3		TSTLIB	00/05/24	17:26:00	TSTLIB	TSTLIB
***** SCLM メンバー・リストの終り *****						

会計情報の表示

TD77.TSTLIB.PLI (TEST1): 会計レコード	
物理データ・セット : TD77I.TSTLIB.PLI	
会計状況	: EDITABLE
変更ユーザー ID.	: Exxxxx
MEMBER バージョン : 1	
言語	: PLI2
作成日付	: 00/05/24
作成時刻	: 17:25:40
プロモート USER ID : Exxxxx	
プロモート日付	: 00/05/24
プロモート時刻	: 17:36:34
先行処理日付	: 00/00/00
先行処理時刻	: 00:00:00
変更グループ	: DEVLIB
許可コード	: PROD
変更許可コード	:
翻訳 PGM バージョン :	
変更日付	: 00/05/24
変更時刻	: 17:25:00
アクセス・キー	:
組立マップ名	:
組立マップ・タイプ :	
組立マップ日付	: 00/05/24
組立マップ時刻	: 17:25:00
"/" を入力してオプション選択	
統計の表示	
変更コードの数	: 0
組み込みの数	: 1
コンパイル単位の数	: 0
ユーザー・エントリーの数	: 0

2.6 ライブラリー・ユーティリティー(オプション3.1)

組立てマップ・レコードの表示

TD77.TSTLIB.PLI (TEST1): 組立マップ・レコード	
一般データ :	
変更 USER ID . . . : EXXXXX	変更グループ . . . : DEVLIB
メンバー版 . . . : 1	変更日付 : 00/05/24
言語 : CCMAP	変更時刻 : 17:31:19
作成日付 : 00/05/24	プロモート日付 . . : 00/05/24
作成時刻 : 17:31:19	プロモート時刻 . . : 17:36:34
	プロモート USERID: EXXXXX
言語バージョン . . . : 1.0	組立マップ日付 . . : 00/05/24
組立マップ名 : TEST1	組立マップ時刻 . . : 17:31:19
組立マップ・タイプ : PLI	
"/" を入力してオプション選択 組立マップ内容の再表示	

組立てマップ内容の表示

BROWSE TD77.TSTLIB.PLI (TEST1): 組立マップ内容		行	00000000
***** データの始め *****			
BUILD MAP CONTENTS			

Keyword	Member	Type	Last Time Modified Ver

SINC	TEST1	PLI	00/05/24 17:25:00 1
OBJ	TEST1	OBJ	00/05/24 17:31:19 1
***** データの終り *****			

2.7 移行ユーティリティ(オプション3.3)

SCLM編集機能を使用せずに、SCLM外で作成されたメンバーを解析し、会計情報を作成

SCLM 移行ユーティリティ - 入力パネル			
選択基準 :			
プロジェクト	:	TD77	
グループ	:	DEVLIB	
タイプ	:	PLI	
メンバー	:	*	(パターンが使用できます。)
メンバー情報 :			
許可コード	:		モード . . . 1 1. 条件付き
変更コード	:		2. 無条件
言語	:	PLI2	3. 強制
出力制御 :			
	EX SUB		処理 . . . 1 1. 実行 (EX)
メッセージ	3 3	1. 端末	2. サブミット (SUB)
リスト	. . 3 3	2. 印刷装置	
		3. データセット	印刷装置 . . .
		4. なし	ボリューム . .

1. 主な入力フィールドの説明

タイプ : 移行の処理対象となるタイプはソースおよびARCHDEF。
組立ての結果得られる出力タイプ(OBJ, LOAD等)に関しては、移行するのではなく、SCLMで組立てることによって生成する。

許可コード : 省略時値を使用するのでブランク

変更コード : 使用しないのブランク

言語 : メンバーに割り当てる「言語識別名」。あらかじめプロジェクト定義で定義してあるものでなければならない。

- 移行処理はメンバー名に*を指定することで、割り当てる言語ごと一括して行える。このとき該当ライブラリーには同じ言語を割り当てたいメンバーだけをコピーしておく。
- すでに会計情報が存在するメンバーに対して、移行処理は行われない。

2.8 アーキテクチャー報告(オプション3.5)

適用業務を構成するソフトウェア要素のリストを作成

SCLM アーキテクチャー報告書ユーティリティー - 入力パネル			
報告書入力：			
プロジェクト	TD77		
グループ	DEVLIB		
タイプ	ARCHDEF		
メンバー	HL@TEST		
報告書カットオフ	 6	
		1. HL	
		2. LEC	
		3. CC	
		4. 総称	
		5. トップ・ソース	
		6. なし	
出力制御：			
	EX SUB	処理 . . . 1	
メッセージ	3 3	1. 実行 (EX)	
報告書	. . 3 3	2. サブミット (SUB)	
		3. データセット	
		4. なし	
		印刷装置	. . . H
		ボリューム	. .

1. 主な入力フィールドの説明

タイプ : ARCHDEFを指定

報告制限 : 報告書の作成基準

指定	報告書に含まれるアーキテクチャー・メンバー
HL	HL
LEC	HL , LEC
CC	HL , LEC , CC
GEN	HL , GEN
TOP	HL , LEC , CC , GEN , ソース
NONE	HL , LEC , CC , GEN , ソース(組み込みを含む)

2.8 アーキテクチャ報告(オプション3.5)

アーキテクチャ報告書の例

```

*****
*****
**
**
**      SOFTWARE CONFIGURATION AND LIBRARY MANAGER (SCLM)      **
**
**      ARCHITECTURE REPORT      **
**
**      00/05/24    20:25:20      **
**
**
**
**      PROJECT:    TD77      **
**      GROUP:      DEVLIB      **
**      TYPE:       ARCHDEF      **
**      MEMBER:     HL@TEST      **
**      CUTOFF:     NONE      **
**
**
*****
*****
1                                     PAGE 2
=====
*
*      ARCHITECTURE REPORT      *
*
*
*  H = HIGH LEVEL      C = COMPILE CONTROL  T = TOP SOURCE E = ERROR  *
*  L = LINKEDIT CONTROL G = GENERIC          I = INCLUDED   D = DEFAULT *
*
=====
CODE:  H   MEMBER:  HL@TEST

---+---1---+---2---+---3---+---4---+---5---+---6---+---7---+---
H HL@TEST  ARCHDEF
L TEST3    ARCHDEF
D TEST3    PLI
T TEST3    PLI
L TEST2    ARCHDEF
D TEST2    PLI
T TEST2    PLI
L TEST1    ARCHDEF
D TEST1    PLI
T TEST1    PLI

```

```

NUMBER OF HIGH LEVEL MEMBERS PROCESSED      =    1
NUMBER OF LINK EDIT CONTROL MEMBERS PROCESSED =    3
NUMBER OF GENERIC MEMBERS PROCESSED          =    0
NUMBER OF DEFAULT MEMBERS PROCESSED          =    3
NUMBER OF COMPILATION CONTROL MEMBERS PROCESSED =    0
NUMBER OF TOP MEMBERS PROCESSED              =    3
NUMBER OF INCLUDED MEMBERS PROCESSED          =    0
NUMBER OF ERROR MEMBERS FOUND                =    0

```

1

PAGE 3

```

=====
*
*          CROSS REFERENCE FOR TYPE:      ARCHDEF
*
=====

```

MEMBER	REF. ARCH. MEM.	TYPE	KEYWORD	INCLUDE-SET
-----	-----	-----	-----	-----
HL@TEST	HL@TEST	ARCHDEF	PROM	
TEST1	TEST1	ARCHDEF	PROM	
	HL@TEST	ARCHDEF	INCL	
TEST2	TEST2	ARCHDEF	PROM	
	HL@TEST	ARCHDEF	INCL	
TEST3	TEST3	ARCHDEF	PROM	
	HL@TEST	ARCHDEF	INCL	

TOTAL MEMBERS PROCESSED FOR TYPE = 7

1

PAGE 4

```

=====
*
*          CROSS REFERENCE FOR TYPE:      LMOD
*
=====

```

MEMBER	REF. ARCH. MEM.	TYPE	KEYWORD	INCLUDE-SET
-----	-----	-----	-----	-----
TEST1	TEST1	ARCHDEF	LOAD	
TEST2	TEST2	ARCHDEF	LOAD	
TEST3	TEST3	ARCHDEF	LOAD	

TOTAL MEMBERS PROCESSED FOR TYPE = 3

1

PAGE 5

```

=====
*
*          CROSS REFERENCE FOR TYPE:      OBJ
*
=====

```

MEMBER	REF. ARCH. MEM.	TYPE	KEYWORD	INCLUDE-SET
-----	-----	-----	-----	-----

TEST1	TEST1	PLI	OBJ
TEST2	TEST2	PLI	OBJ
TEST3	TEST3	PLI	OBJ

TOTAL MEMBERS PROCESSED FOR TYPE = 3

1

PAGE 6

```
=====
*
*          CROSS REFERENCE FOR TYPE:      PLI
*
=====
```

MEMBER	REF. ARCH. MEM.	TYPE	KEYWORD	INCLUDE-SET
-----	-----	-----	-----	-----
TEST1	TEST1	PLI	SINC	
	TEST1	PLI	PROM	
	TEST1	ARCHDEF	INCLD	
TEST2	TEST2	PLI	SINC	
	TEST2	PLI	PROM	
	TEST2	ARCHDEF	INCLD	
TEST3	TEST3	PLI	SINC	
	TEST3	PLI	PROM	
	TEST3	ARCHDEF	INCLD	

TOTAL MEMBERS PROCESSED FOR TYPE = 9

2.9 監査及びバージョン(オプション3.8)

SCLM トランザクションについての監査、バージョン付け、会計情報を提供し、任意選択プロジェクト階層内のSCLM制御のバージョンまたは複写を保守

SCLM 監査およびバージョン・ユーティリティ - 入力パネル		続く
オプション . . . 1	1. バージョン、および監査追跡 2. バージョンのみ	
SCLM ライブラリー :		
プロジェクト	TD77	
グループ . . .	TSTLIB	
タイプ . . .	PLI	
メンバー . . .	(メンバー・リストにはメンバー名、またはブランク)	
選択日付の範囲 :		
開始日付 . . .	(開始日付、またはブランク)	
終了日付 . . .	(終了日付、またはブランク)	
他の非 SCLM 制御検索データ・セット :		
データ・セット名	VERSION	
SCLM 検索グループおよびタイプ :		
グループ . . .	許可コード . . .	
コマンド	====>	

1. 主な入力フィールドの説明

グループ : 監査情報とバージョン情報が必要なグループ。
プロジェクト定義で監査データセットが定義されものを指定。また、そのグループはプロジェクト定義のFLMATVERマクロでも定義されていなければならない。

タイプ : メンバーを含むタイプを指定。(必須フィールド)
プロジェクト定義のFLMATVERマクロで定義されているものを指定。

データセット名 : 検索されたメンバーのバージョンが格納される順次データセットの名前。
DCD属性のRECFM及びLRECLは、作成元のデータセットの属性と一致しなければならない。

注) 監査制御データセット(VSAM)と区分バージョンデータセットに保管されているデータは、監査バージョン・ユーティリティ専用。

監査バージョン・ユーティリティを使用せずに、これらのデータセットの編集または変更を行わないこと。これを行うとデータが失われる。

2.9 監査及びバージョン(オプション3.8)

SCLM - バージョン選 行 1,076 - 1,078 / 1,078									
プロジェクト . . . : TD77									
タイプ : PLI									
データ・セット . . : EXXXXX.VERSION									
行コマンド : A 監査情報 D 削除 R 検索									
S	メン バー	グル ープ	処置 理由	処置 日付	処置 時刻	ユーザ ID	処置 結果	VER	状況

	TEST1	TSTLIB	PROMOTE	00/05/24	17:36:42	EXXXXX	COMPLETE	*	
	TEST2	TSTLIB	PROMOTE	00/05/24	17:37:51	EXXXXX	COMPLETE	*	
	TEST3	TSTLIB	PROMOTE	00/05/24	17:37:55	EXXXXX	COMPLETE	*	
***** データの終り *****									

SCLM - 監査 / バージョン・レコード			
プロジェクト : TD77			
監査データ :			
グループ	TSTLIB	呼出しサービス . . .	PROMOTE
タイプ	PLI	対応処置	PUT
メンバー	TEST1	処置結果	COMPLETE
監査日付	00/05/24	失敗メッセージ . . .	
監査時刻	17:36:42.74		
ユーザー ID	EXXXXX		
SCLM 変更日付 . . .	00/05/24		
SCLM 変更時刻 . . .	17:25:00		
バージョン・データ :			
データ・セット . . .	<u>TD771.TSTLIB.SOURCE.VERSION</u>		
メンバー	<u>TEST1</u>	要求形式	DELTA
変更日付	00/05/24	現行形式	DELTA
変更時刻	17:37:51		
"/" を入力してオプション選択 ;			
会計情報の表示			

第3章 アーキテクチャー定義

- 3.1 アーキテクチャー定義
- 3.2 CC(コンパイル) アーキテクチャー・メンバー
- 3.3 LEC(関係編集制御) アーキテクチャー・メンバー
- 3.4 HL(ハイ・レベル) アーキテクチャー・メンバー
- 3.5 Generic(汎用) アーキテクチャー・メンバー
- 3.6 アーキテクチャー定義コーディング・ガイドライン
- 3.7 アーキテクチャー定義 まとめ

3.1 アーキテクチャー定義

1. ソフトウェアの組立てに必要な情報を S C L M に定義

- 組立ての入力（コンパイルされるソース、関係編集されるオブジェクト、ロード・・・）
- 組立ての出力（出力されるオブジェクト、ロード、Listect .. の名前、保管場所）
- 組立て時のオプション（コンパイラ・オプション、関係編集オプション...）

2. アーキテクチャー・メンバーの4つの種類

- CC （コンパイル制御） コンパイルの入力（ソース）と出力（OBJ）の指定
- LEC （関係編集制御） 関係編集の入力（OBJ，LOAD）と出力（LOAD）の指定
- HL （上位） 複数の構成要素をまとめて「適用業務」を構成
- Generic （汎用） コンパイルや関係編集以外の一般的な変換処理用

これら4つをSCLMに含まれるステートメントの内容によって区別する。

明示的にどのタイプであるかの宣言等はない。

3. アーキテクチャー・ステートメントのコーディング

- 80バイト、固定長ブロック化（FB）←ARCHDEFデータセット
- 有効桁は1～72バイト、継続不可
- 大文字、小文字どちらでも可、ブランクは何文字分あってもよい（注）
- SCLMは制御ステートメント、パラメータの妥当性は検査しない
- 1桁目が*の行はコメントとして扱われる

（注）CMDステートメントはブランク、大文字/小文字すべて意味を持つ

アーキテクチャー定義の例（LEC）

```
*****
* MEMBER NAME      :  MODULE1                      *
* DEFINITION TYPE  :  LEC                          *
* DESCRIPTION      :  LEC ARCHITECTURE FOR BATCH EXECUTABLE LOAD *
*                  :  MODULE                        *
*****
LKED    LINK2                      * LANGUAGE NAME FOR LINK/EDIT *
LOAD    MODULE1    LMOD            * LOAD MODULE / LIB TYPE(LMOD) *
INCLD   MODULE1    PLI             * SOURCE NAME / LIB TYPE(PLI)  *
PARM    AMODE(24),RMODE(24)
```

3.2 CC-コンパイル制御アーキテクチャー・メンバー

CC:

ソース・コードからオブジェクト・モジュールを作成する場合に定義

但し、他の方法で代替可能なため一般的にCCは作成されないことが多い

1. コンパイル対象ソース・メンバーの指定 (SINCステートメント)

SINC ソース・メンバー名 タイプ名

- 必須ステートメント(少なくとも1つ必要)
- CCでは他のアーキテクチャー・メンバーは参照できない(ソースのみを参照)
- INCLUDEされるメンバーはSCLMが自動的に組み込むので指定不要
- 呼び出されるコンパイラはソース・メンバーに関連付けられた言語定義で決定

2. オブジェクト・モジュールの指定 (OBJステートメント)

OBJ オブジェクト・モジュール名 タイプ名

- このステートメントはCCにのみ指定可能

3. コンパイラ・オプションの指定 (PARMステートメント)

PARM PARM1 , PARM2 , PARM3 ,

- 言語定義 (FLMTRNSL) の省略時オプションを変更する場合
- オプション指定の可否はプロジェクト定義、言語定義のパラメータで制御
- PARMステートメントにはコメントを続けない

CC アーキテクチャー定義の例

```
*****
* MEMBER NAME      :  MODULE1C                                *
* DEFINITION TYPE   :  CC                                      *
* DESCRIPTION       :  CC ARCHITECTURE FOR MODULE1            *
*
*****
OBJ    MODULE1     OBJ                                * MEMBER NAME / LIB TYPE(OBJ) *
SINC   MODULE1     PLI                                * MEMBER NAME / LIB TYPE(SOURCE) *
PARM   OPT(0) , TEST(ALL,SYM)
```

48

3.3 LEC - 連係編集制御アーキテクチャー・メンバー

LEC:

オブジェクト・モジュール、ロード・モジュールを連係編集してロード・モジュールを作成する場合に定義 (LECは必ず作成しなければならない)

1. 連係編集の対象の指定 (INCLD , INCLステートメント)

INCLD	ソース・メンバー名	タイプ名
INCL	アーキテクチャー・メンバー名	タイプ名

- (1) 必須ステートメント(少なくとも1つ以上必要)
- (2) ソースを直接参照する場合はINCLDを指定 (CCを省略するケース)
- (3) 他のアーキテクチャー・メンバー (CC、LEC) を参照する場合はINCELを指定
- (4) HL , Genericの参照はできない
- (5) ソースまたはCCを参照する場合はコンパイルされた結果のオブジェクト・モジュールが組み込まれる
- (6) LECを参照する場合はロード・モジュールが組み込まれる

2. 一緒に連係編集されるロード・モジュールの指定 (INCL , LINKステートメント)

INCL	アーキテクチャー・メンバー名	タイプ名
LIND	ロード・モジュール名	タイプ名

- (1) INCLステートメントで他のLECを参照 (そのモジュールが最新かを検査)
- (2) LINKステートメントを指定 (検査を行わずに組み込む) 普通は使用しない

3. 出力されるロード・モジュールの指定 (LOADステートメント)

LOAD	ロード・モジュール名	タイプ名
------	------------	------

LOADステートメントはLECでのみ指定可能

4. 連係編集オプションの指定 (PARM , PARMxステートメント)

PARM PARM1 , PARM2 , PARM3 ,

3.3 LEC - 連係編集制御アーキテクチャー・メンバー

5. 連係編集制御ステートメントの指定(CMDステートメント)

CMD 連係編集制御ステートメント

- オブジェクト・モジュール、ロード・モジュールとともに連係編集プログラムに
- 渡される制御ステートメントを指定(指定位置に注意)

(例) CMD INCLUDE SYSLIB(XXXX)

6. ロード・モジュールの別名を指定 (ALIASステートメント)

ALIAS ALIAS名 タイプ名

- ALIASステートメントを使用せずに、CMDステートメントでALIASを指定しても、作成されたALIASはSCLMを認識されないため、プロモートされないことに注意

7. 呼び出す連係編集プログラムの指定(LKEDステートメント)

LKED 言語識別名

- プロジェクト定義に事前に定義してある言語識別名を指定
- SYSLIBに連結してあるライブラリーの組み合わせごとに、いくつかの異なる言語を用意してあるような場合に使用

アーキテクチャー定義の例 (LEC)

```
*****
* MEMBER NAME      :  MODULE1                      *
* DEFINITION TYPE  :  LEC                          *
* DESCRIPTION      :  LEC ARCHITECTURE FOR BATCH EXECUTABLE LOAD *
*                  :  MODULE                        *
*****
LKED    LINK2                      * LANGUAGE NAME FOR LINK/EDIT *
LOAD    MODULE1    LMOD            * LOAD MODULE / LIB TYPE(LMOD) *
INCLD   MODULE1    PLI             * SOURCE NAME / LIB TYPE(PLI)  *
PARM    AMODE(24),RMODE(24)
```

3.4 HL-ハイ・レベル・アーキテクチャー・メンバー

HL:

他のアーキテクチャー・メンバーをまとめて「適用業務」を定義する場合に使用

- 複数のHLを階層的に使用することで部分適用業務を定義していき、最終的に適用業務全体を1つのHLで定義できる（1つのHLですべての構成要素を制御）
- BUILD, PROMOTEに適用業務全体を表すHLを使用すると「適用業務全体の完全性」が保証される

1. 他のアーキテクチャー・メンバーの参照（INCLステートメント）

INCL アーキテクチャー・メンバー名 タイプ名

- 階層的にアーキテクチャー・メンバーをネストさせることが可能
- 適用業務全体を最上位の1個のHLアーキテクチャー・メンバーで制御可能
- HL及びLECを参照可能(CC、Genericは参照できない)

2. コンパイラー等で処理されないソフトウェア要素の指定(PROMステートメント)

PROM メンバー名 タイプ名

- パネル、メッセージ、テキストなどコンパイル/リンクされないソースに使用
- 関連するプログラム・ソースなどとの「データ上の依存関係」を定義
- PROMステートメントにより、アーキテクチャー・メンバーで処理されるモジュールと共に、プロモート（だけ）されるメンバーを指定する。
（指定されたメンバーはコンパイル/リンク等の組立て処理は行われない）
- PROMステートメントの後にはコメントを続けない

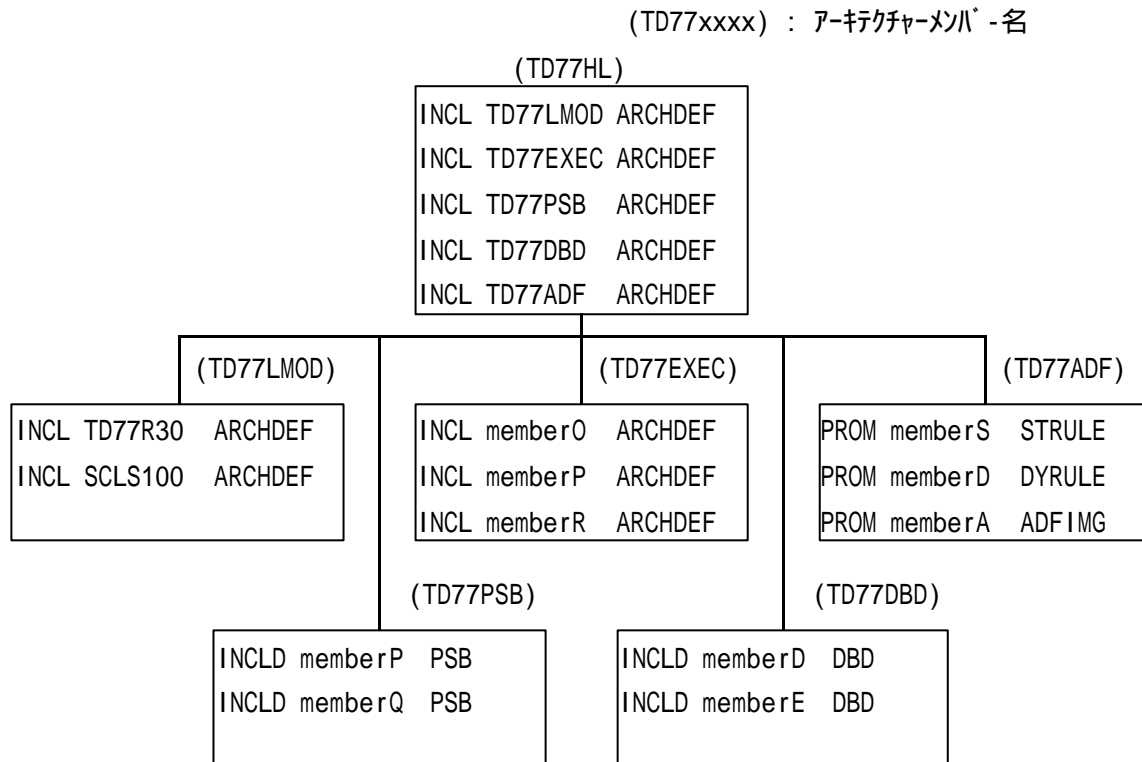
HLアーキテクチャー定義の例

```
*****
* MEMBER NAME      : PROJ01HL                                *
* DEFINITION TYPE   : HL                                      *
* DESCRIPTION       : HL ARCHITECTURE FOR WHOLE APPLICATION  *
*****
INCL    HL@SUB1     ARCHDEF          * MEMBER NAME / TYPE(ARCHDEF) *
INCL    HL@SUB2    ARCHDEF          * MEMBER NAME / TYPE(ARCHEEF) *
```

3.4 HL-ハイ・レベル・アーキテクチャー・メンバー

3.HLアーキテクチャー・メンバーのAMSサービスでの使用例

アプリケーション(TD77)におけるHLアーキテクチャー定義の構造



4.PROMステートメントのAMS サービスでの使用例

PROMステートメントは「組立て」されないタイプに使用され、AMSサービスではADFのソース管理orソース(マクロ)のみ管理する場合に使用している。

(注)ADFの仕組みはSCLMの「組立て」として実現するのが難しいためにADF関連ソースは、組立てされない「テキスト・タイプ」として扱っている。

```

*****
* MEMBER NAME      :  ppppADF                                *
* DEFINITION TYPE  :  HL                                      *
* DESCRIPTION      :  HL ARCHITECTURE FOR ADF SOURCES        *
*****

PROM  xxxxxxS      STRULE
PROM  xxxxxxD      DYRULE
PROM  xxxxxxA      ADFIMG
  
```


3.5 Generic - 汎用アーキテクチャー・メンバー

Generic:

コンパイラ、関係編集プログラム以外の処理プログラムによって生成される汎用出力を定義

- 基本的な特徴はCCと同じ(OBJステートメント以外は共通)
- オブジェクト・モジュール、ロード・モジュールを生成しない処理プログラム
- 1つ言語定義でコンパイルとリンクを一度に行う場合(ソースからロード・モジュールを生成)
- (例) SCRIPT/VS処理プログラムの場合
ドキュメント・ソース(SCRIPTSOURCE)からフォーマットされた形式の出力(LIST38XX等)を出力

1. 汎用出力の指定 (OUTxステートメント)

OUTx メンバー名 タイプ名 (但しxは0~9)

- OBJ, LOAD, LIST, LMAP以外の出力を定義

Genericアーキテクチャー定義の例

```
*****
* MEMBER NAME      :  Gen@SC1                               *
* DEFINITION TYPE   :  Generic                               *
* DESCRIPTION       :  Generic ARCHITECTURE FOR SCRIPT1     *
*****
SINC      SCRIPT1                                * MEMBER NAME / LIB(SCRIP) *
OUT1      SCRIPT1                                * MEMBER NAME / LIB(LIST3820) *
```

3.6 アーキテクチャー定義コーディング・ガイドライン

アーキテクチャー定義の作成に当たっては以下のルールに従う

- LEC アーキテクチャー定義のメンバー名は、原則としてモジュール名と同じにする
(CCを作成する場合は、サフィックス“C”をつける)
- HL アーキテクチャー定義はタイプごとに作成し、メンバー名は以下のようにする

ppppLMOD	バッチ・ロード・モジュール
ppppEXEC	オンライン・ロード・モジュール
ppppDBD	DBD
ppppPSB	PSB

- 次の情報を含むコメント・ブロックを先頭にコーディングする

```

メンバー名
アーキテクチャー定義タイプ(HL/LEC/CC/Generic)
メンバー記述

```

- ステートメントは1桁目より開始する
- メンバー名、識別子などは10桁目より開始する
- その他のパラメータ(ライブラリーのタイプなど)は20桁目より開始する
- 各行に40桁目よりコメントを記述する(但しコメントを指定すると意味が変わってしまうステートメントにはつけない。CMD,PARM,PROM)

アーキテクチャー定義のコーディング例

```

*****
* MEMBER NAME       :  ppppXXX                               *
* DEFINITION TYPE   :  LEC                                   *
* DESCRIPTION       :  LEC ARCHITECTURE FOR MODULE ppppXXX   *
*****
LKED   LINK2                               * LANGUAGE NAME FOR LINK/EDIT
LOAD   ppppXXX  LMOD                       * MEMBER NAME / LIB TYPE (LOAD)
INCLD   ppppXXX  PLI                       * MEMBER NAME / LIB TYPE (SOURCE)
PARM    AMODE(24),RMODE(24)

```

3.7 アーキテクチャー定義 まとめ

1. 組立て(Build) に対して入力と出力を定義

- ロード・モジュール1個につき1個のLECアーキテクチャーを作成
(CCはコンパイル・オプションの変更が必要な場合以外は作成しなくてもよい)
- モジュール全体に対して1つのHLアーキテクチャーを作成
(必要に応じてHLを階層化してサブ・アプリケーションを定義)
- 定義はタイプ ARCHDEF 保存され他のメンバーと同様SCLMによって管理される

2. アーキテクチャー定義のタイプと有効なステートメント

CC	LEC	HL	Generic
CMD	ALIAS	COPY	CMD
COPY	CMD	INCL	COPY
LIST	COPY	INCLD	LIST
OBJ	INCL	PROM	OUTx
OUTx	INCLD		PARM
PARM	LINK		PARM x
PROM	L MAP		PROM
SINE	LOAD		SINE
	OUTx		
	PARM		
	PARM x		
	PROM		

第4章 プロジェクト定義

4.1 プロジェクト定義

4.2 代替プロジェクト

4.1 プロジェクト定義

1. プロジェクトの構造、特性をSCLMに定義

- プロジェクトの識別名 (データセットの第1修飾子)
- プロジェクト・データセットの階層構造、グループ、タイプ
- その他プロジェクト固有の制御オプション (会計DS名、バージョンDS名)

2. プロジェクト定義データセット

- アセンブル、リンクされたプロジェクト定義が格納される PDS
- pppp.PROJDEFS.LOADの名前を持つ (名前は固定)
- プロジェクト定義はプロジェクト名に等しいメンバー名を持つ (基本プロジェクト)
(例) TD77.PROJDEFS.LOAD(TD77)
- 基本プロジェクトに加え、任意の数の「代替プロジェクト」を持つことができる
- プロジェクト定義データセットが存在するとISPF/PDFの編集機能は同じプロジェクト名で始まるデータセットを (たとえそれがSCLM管理のデータセットでなくても) 編集する際、警告メッセージを出す。

==> 編集中のメンバーはSCLMに制御されているプロジェクトに属する。

3. プロジェクト定義のコーディング

- プロジェクト定義はS/370のアセンブラーマクロで記述
- 有効桁は1～71桁で継続する場合は、72桁目に継続文字('C')を指定し、次行を16桁目から開始する
- 必須パラメータに無効な値を指定するとRC=8でエラーとなる
- 任意パラメータに無効な値を指定するとRC=4で省略時値が使用される
- SCLMはアセンブリー時には指定されたデータセット妥当性検査は行わない

4.1 プロジェクト定義

4. プロジェクト定義で使用する主要なマクロ

- | | |
|--------------|--|
| (1) FLMALEG | プロジェクト識別子 (データセットの第一修飾子) |
| (2) FLMALTC | 指定されたグループ固有の制御オプション
(FLMCNTRLで指定するものとは異なる制御情報を指定できる。
～FLMCNTRLマクロからの値を使用しない～
FLMGROUPマクロのALTCパラメータで参照される) |
| (3) FLMGROUP | グループ (第2修飾子) |
| (4) FLMTYPE | タイプ (第3修飾子) |
| (5) FLMATVER | 監査・バージョンユーティリティーの制御オプション
(そのユーティリティーで獲得する情報のタイプ・活動状態を
定義～FLMALTC,FLMCNTRL参照～) |
| (6) FLMCNTRL | プロジェクト固有の制御オプション
(会計・監査制御・区分バージョンのD/S名、最大VIO値、コン
パイラーオプション指定の可否...) |
| (7) FLMAEND | プロジェクト定義の終わり |

プロジェクト定義の例 ～ TD77I.MODEL.SOURCE(TD75,TD75M)を参照～

```

TITLE '*** PROJECT DEFINITION FOR PROJECT=TD75 ***'
*
* *****
*      *   DEFINE THE PROJECT NAME   *
*      *****
TD75  FLMABEG  <-- プロジェクト識別子 (データセットの第一修飾子になる) +-- プロジェ
クト名 (アプリケーション名)
*
* *****
*      *   DEFINE THE ALTERNATIVE ACCOUNT D/S & FLEXIBLE NAMING   *
*      *****
RELLIBC  FLMALTC  ACCT=TD75IV.ACCOUNT.FILE,                                C
           DSNAM=TD75.@@FLMGRP.@@FLMTYP
UTLLIBC  FLMALTC  ACCT=TD75IVT.ACCOUNT.FILE,                               C
           DSNAM=TD75I.@@FLMGRP.@@FLMTYP,                                C
           VERS=TD75IVT.AUDIT.FILE,                                       C
           VERPDS=TD75I.UTLLIB.SOURCE.VERSION

```

```

* *****
*      *      DEFINE THE GROUPS      *
*      *****
*      +- - グループ (第2修飾子)
RELLIB  FLMGROUP AC=(PROD),KEY=Y,ALTC=RELLIB
TSTLIB  FLMGROUP AC=(PROD),PROMOTE=RELLIB,KEY=Y
DEVLIB1 FLMGROUP AC=(PROD),PROMOTE=TSTLIB,KEY=Y
DEVLIB2 FLMGROUP AC=(PROD),PROMOTE=TSTLIB,KEY=Y
UTLLIB  FLMGROUP AC=(TEST),PROMOTE=RELLIB,KEY=Y,ALTC=UTLLIB
*
*      *****
*      *      DEFINE THE TYPES      *
*      *****
*      +- - グループ (第3修飾子)   ここで必要なタイプを指定する。
ASM      FLMTYPE EXTEND=MAC
MAC      FLMTYPE
PLI      FLMTYPE EXTEND=PLINCL
RULES    FLMTYPE EXTEND=PLINCL
PLINCL   FLMTYPE
C        FLMTYPE EXTEND=CINCL
CINCL    FLMTYPE
OBJ      FLMTYPE
PLKOBJ   FLMTYPE
NCAL     FLMTYPE
EXEC     FLMTYPE
EXEC2    FLMTYPE
LMOD     FLMTYPE
ISPLLIB  FLMTYPE
PSB      FLMTYPE
PSBOBJ   FLMTYPE
PSBLIB   FLMTYPE EXTEND=DBDLIB
DBD      FLMTYPE
DBDLIB   FLMTYPE
ARCHNCL  FLMTYPE
ARCHDEF  FLMTYPE
ARCHPLK  FLMTYPE
DBRM     FLMTYPE
STRULE   FLMTYPE
DYRULE   FLMTYPE
ADFIMG   FLMTYPE
*
*      *****
*      *      PROJECT CONTROLS      *
*      *****
+- - 監査、バージョン・ユーティリティーを使用可能にする。対象となるグループ、タイプを指定。
FLMATVER GROUP=TSTLIB,TYPE=PLI,VERSION=YES
FLMATVER GROUP=TSTLIB,TYPE=PLINCL,VERSION=YES
FLMATVER GROUP=TSTLIB,TYPE=ASM,VERSION=YES
FLMATVER GROUP=TSTLIB,TYPE=MAC,VERSION=YES
FLMATVER GROUP=TSTLIB,TYPE=C,VERSION=YES
FLMATVER GROUP=TSTLIB,TYPE=CINCL,VERSION=YES
FLMATVER GROUP=TSTLIB,TYPE=RULES,VERSION=YES

```

```

FLMATVER GROUP=TSTLIB,TYPE=PSB,VERSION=YES
FLMATVER GROUP=TSTLIB,TYPE=DBD,VERSION=YES
FLMATVER GROUP=TSTLIB,TYPE=STRULE,VERSION=YES
FLMATVER GROUP=TSTLIB,TYPE=DYRULE,VERSION=YES
FLMATVER GROUP=TSTLIB,TYPE=ADFIG,VERSION=YES
FLMATVER GROUP=UTLLIB,TYPE=PLI,VERSION=YES
FLMATVER GROUP=UTLLIB,TYPE=PLINCL,VERSION=YES
FLMATVER GROUP=UTLLIB,TYPE=ASM,VERSION=YES
FLMATVER GROUP=UTLLIB,TYPE=MAC,VERSION=YES
FLMATVER GROUP=UTLLIB,TYPE=C,VERSION=YES
FLMATVER GROUP=UTLLIB,TYPE=CINCL,VERSION=YES
FLMATVER GROUP=UTLLIB,TYPE=RULES,VERSION=YES
FLMATVER GROUP=UTLLIB,TYPE=PSB,VERSION=YES
FLMATVER GROUP=UTLLIB,TYPE=DBD,VERSION=YES
FLMATVER GROUP=UTLLIB,TYPE=STRULE,VERSION=YES
FLMATVER GROUP=UTLLIB,TYPE=DYRULE,VERSION=YES
FLMATVER GROUP=UTLLIB,TYPE=ADFIG,VERSION=YES

```

+- プロジェクト固有の制御オプションの指定

```

FLMCNTRL ACCT=TD75IVT.ACCOUNT.FILE,          C
          DSNAME=TD75I.@@FLMGRP.@@FLMTYP,      C
          VERS=TD75IVT.AUDIT.FILE,             C
          VERPDS=TD75I.TSTLIB.SOURCE.VERSION,  C
          MAXVIO=1,                             C
          VIOUNIT=SYSDA,                         C
          OPTOVER=Y,                             C
          DASDUNIT=TSOUNIT1

```

*

*

*

*

*

*

*

*

*

*

*

*

*

*

*

*

*

*

*

*

*

*

*

*

*

*

*

*

*

*

*

*

*

PARSERS

COPY FLM@ARCD

-- ARCHITECTURE DEF. LANGUAGE

PARSERS + TRANSLATOR 以下で必要な言語定義メンバーを指定する。

```

COPY @ASM          -- ASM ASSEMBLE
COPY @C370         -- C370 COMPILE
COPY @C370C        -- CICS-C370 COMPILE
COPY @C370DB2      -- C370-DB2 COMPILE
COPY @C370CDB2     -- CICS-C370-DB2 COMPILE
COPY @RULE1        -- RULES COMPILE PLI V1
COPY @RULES        -- RULES COMPILE PLI V2
COPY @RULEDB1      -- RULES-DB2 COMPILE PLI V1
COPY @RULEDB2      -- RULES-DB2 COMPILE PLI V2
COPY @PLI1         -- PL/I COMPILE + CMPAT(V1)
COPY @PLI2         -- PL/I COMPILE + CMPAT(V2)
COPY @PLIC         -- CICS-PL/I COMPILE
COPY @PLIC3        -- CICS-PL/I COMPILE (V3.2)
COPY @PLI1DB2      -- PL/I-DB2 COMPILE + CMPAT(V1)
COPY @PLI2DB2      -- PL/I-DB2 COMPILE + CMPAT(V2)
COPY @PLICDB2      -- CICS-PL/I-DB2 COMPILE

```



```
COPY @PLC3DB2          -- CICS-PL/I-DB2 COMPILE (V3.2)
COPY @PLINK             -- LINKAGE EDIT PRELINK OF C370
COPY @LINKN            -- LINKAGE EDITOR NCAL
COPY @LINK1            -- LINKAGE EDITOR FOR BATCH  V1
COPY @LINK2            -- LINKAGE EDITOR FOR BATCH  V2
COPY @DB2LNK1          -- LINK EDIT FOR DB2 V1(NO-IMS)
COPY @DB2LNK2          -- LINK EDIT FOR DB2 V2(NO-IMS)
COPY @SDTMLN1          -- LINKAGE EDITOR PLI  V1
COPY @SDTMLNK          -- LINKAGE EDITOR PLI  V2
COPY @PSBLNK           -- IMS PSB LINK
COPY @PSB              -- IMS PSBGEN
COPY @PSBOBJ           -- IMS PSB ASSEMBLE
COPY @DBD              -- IMS DBDGEN
COPY @STRULE           -- ADF STATIC RULE
COPY @DYRULE           -- ADF DYNAMIC RULE
COPY @ADFIMG           -- ADF SCREEN IMAGE
COPY @TEXT             -- TEXT TYPE  IMAGE
```

*

*

*

*

```
*****
*      END THE PROJECT DEFINITION      *
*****
```

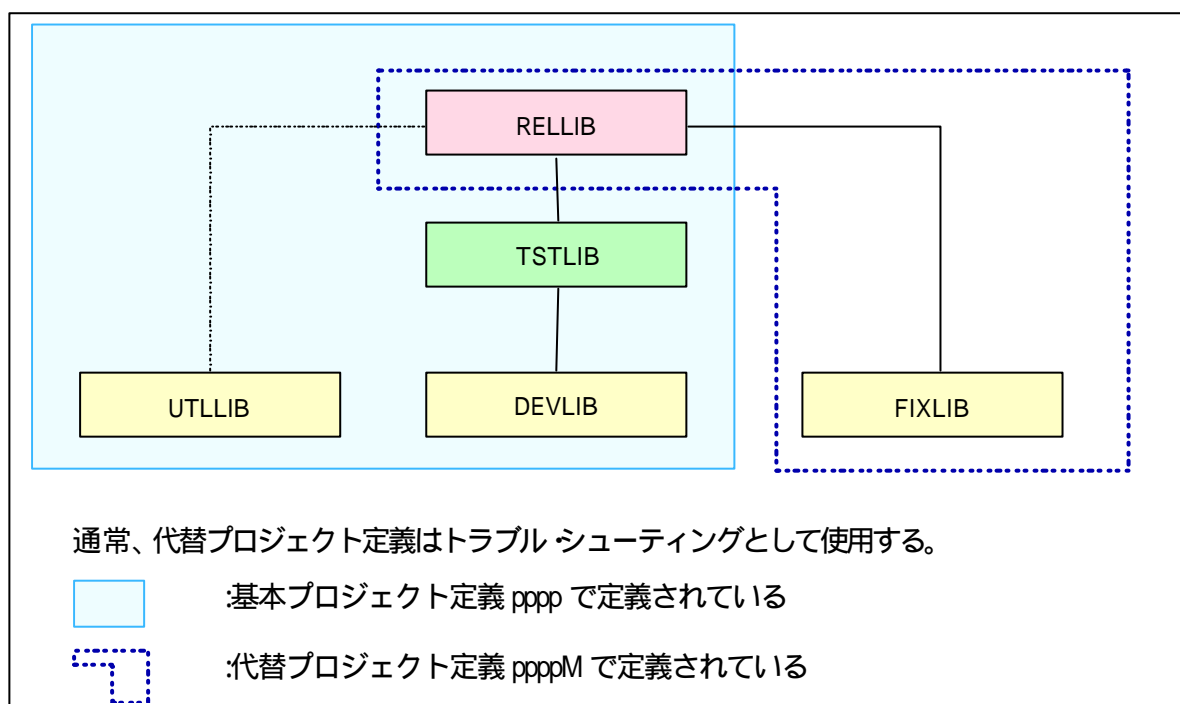
4.2 代替プロジェクト

1つのプロジェクトに対して複数のプロジェクト定義の作成が可能

- 異なるプロジェクト・データセットの構造、グループ、言語等のサポート
- 代替プロジェクトの使用により完全性が損なわれる場合があるので使用には注意が必要

AMSサービスにおける代替プロジェクトの使用例

(プロジェクト名：pppp～プロジェクト定義のFLMABEGで指定)



1. 代替プロジェクトの使用による不整合

- 基本プロジェクトではDEV LIBでPGM1の新リリース用の拡張子作業をおこなっている
- 現在稼働中のPGM1にトラブルが発生したため、代替プロジェクト「ppppM」を指定し、RE LIBからFIX LIBにダウンロードしてトラブルに対する修正を行う

この時PGM1はDEV LIBで変更中であるが、FIX LIBでの編集についてはロックがかからないことに注意。(異なるプロジェクト定義間ではロックされない)

- FIX LIBで変更されたPGM1はRE LIBにプロモートされる。ただしDEV LIBで開発中のPGM1にはFIX LIBで行った変更は反映されない。

4.2 代替プロジェクト

代替プロジェクトの使用による不整合 (続き)

- DEVLIBでの開発が完了し上位層 (TSTLIB, RELLIB) にプロモートを行うとすると、RELLIBのPGM1が、DEVLIBに当初ダウンロードしたときのバージョンと異なるために次のエラーとなる。

```
FLM5319 - WARNING , PREDECESSOR VERIFICATION FAILED
          INPUT GROUP :  DEVLIB  TYPE : SOURCE  MEMBER : PGM1
          ERRORGROUP1 :  DEVLIB  DATE : 92/09/18  TIME : 09 : 09 : 00
          ERRORGROUP2 :  RELLIB  DATE : 92/09/20  TIME : 20 : 07 : 12
```

このメッセージは現在DEVLIBにあるメンバーPGM1の会計情報にある「先祖」のDATE/TIMEフィールドとRELLIBのPGM1の最終修正DATE/TIMEが一致していないことを警告している。

Unconditionalモードでプロモートを行うと、上記の状況においてもプロモートが実施されてしまうことに注意

ユーザーはこの状態を回復するのに以下の手順を実施する

- a) DEVLIBにあるPGM1をSCLM管理外のデータセットへコピーしておく。
- b) SCLMライブラリー・ユーティリティー (3.1) でDEVLIBからPGM1を削除する。この時メンバーと共に会計情報、組立てマップも削除される。
- c) 再度RELLIBからDEVLIBにPGM1をダウンロードして、ステップa) でコピーしておいたPGM1をもとに必要な変更を行う。
- d) DEVLIBから上位層へプロモートを行う。

第5章 言語定義

5.1 言語定義

5.1 言語定義

1. 特定の言語の翻訳手順をSCLMに定義

- コンパイル用のJCLのカタブロに相当
- 使用するコンパイラ、ALLOCATEするDD名、DS名等の指定
- メンバー解析用のPARSERの指定
- 異なる翻訳手順ごとに作成（例：PLI+DB2, PLI+CICS ...）
- 定義した内容はプロジェクト定義と一緒にアセンブル、リンクされる
- 定義した「言語」はSCLM制御下の各ソース・メンバーに関連付けされ、組立て（BUILD）により定義された翻訳手順が実行される（コンパイル）

2. 言語定義の修正（「* APPENDIX * 付3」参照）

- 言語定義をカスタマイズする必要がある場合は以下の内容に限って行う
- 適用業務固有のSYSLIBの追加（SCLM管理外のメンバーの組み込み）では次の2か所データセット名を追加する

1. FLMSYSLBマクロで対象D/Sの指定（一番先頭で記述するマクロ）

（例）PLI2 FLMSYSLB SYS1.DCL
 FLMSYSLB SYS1.UCTRL
 FLMSYSLB PROJ01.DS.SYSLIB

2. 該当するDD名（例えばSYSLIB）に対するFLMALLOCマクロの後にFLMCPYLBマクロで対象DSの指定

（例）FLMALLOC IOTYPE=I, KEYREF=SINC, DDNAME=SYSLIB
 FLMCPYLB SYS1.DCL
 FLMCPYLB SYS1.UCTRL
 FLMCPYLB PROJ01.DS.SYSLIB

SCLMが組み込み関係（INCLUDE）を処理する時、FLMSYSLBで指定したデータセット中にメンバーを見つけると、SCLMはそのメンバーを追跡しない。

（SCLM管理外として扱う。変更があっても関知しない。会計情報を持たない。）

5.1 言語定義

3. その他変更の可能性のあるオプション（特に必要な場合以外は変更しない）

- コンパイラーのオプション（FLMTRNSLのOPTIONSパラメーター）
 - コンパイラーからの戻りコードの許容限度（FLMTRNSLのGOODRCパラメーター）
 - コンパイラーのデータセット名（FLMTRNSLのD/SNAMEパラメーター）
 - 作業データセットのスペース量（FLMALLOCのRECNUMパラメーター）
 - 割り振られるデータセットの属性（FLMALLOCのRECFM, LRECL, BLKSZEパラメーター）
 - プロジェクトで定義された出力タイプ（例えばLMOD）を省略時のタイプとして指定（FLMALLOCのDELTTYPパラメーター）
 - 言語のバージョン（FLMLANGLのVERSIONパラメーター）
- このパラメータを変更するとすべてのメンバーが再コンパイルされる（要注意）

言語定義の例

```
***** 00010000
* LANGUAGE DEFINITION : PLI COMPILE FOR PLI FOR MVS V1.1 00030000
***** 00030000
* +---SCLM管理外のデータセットを定義 00040000
PLIM FLMSYSLB SA00.D/S.DCL 00050000
      FLMSYSLB SYS1.CTRL2 00050000
      FLMSYSLB SYS1.DCL 00050000
      FLMSYSLB SYS1.CTRL 00060000
      FLMSYSLB SYS1.UCTRL 00060000
<--- アプリケーション独自のMACROライブラリーまたはINCLUDEがあればここに追加する。
* 00070000
      FLMLANGL LANG=PLIM,VERSION=1.1
      +---言語のバージョン(ここを変えると全てのメンバーが再コンパイルされる)
* 00090000
*****
* PL/I : LANGUAGE PARSE 00020000
*****
* +---JCL EXECステートメントと同様の機能を定義 00050000
      FLMTRNSL CALLNAM='SCLM PL/I PARSE', 00080000
              FUNCTN=PARSE, 00090000
              COMPILE=FLMLPGEN, 00100000
              PORDER=1, 00110000
              OPTIONS=(SOURCEDD=SOURCE, C 00130000
              PARSEMEM=@@FLMMBR, 00140000
              STATINFO=@@FLMSTP, 00150000
              LISTINFO=@@FLMLIS, 00160000
              LISTSIZE=@@FLMSIZ, 00170000
              LANG=I)
```

```

*          (* SOURCE          *)                                00180000
*          FLMALLOC  IOTYPE=A,DDNAME=SOURCE                    00190000
*          FLMCPYLB  @@FLMD/SN(@@FLMMBR)                        200000
*                                                                00210000
*                                                                00230000
*=====
* STEP 1 :  CRD EXPAND                                         00300000
*=====
*                                                                00290000
*          -- CRD EXPAND --                                     00300000
*          FLMTRNSL  CALLNAM='CRD/EXPAND',                       C00310000
*                   FUNCTN=BUILD,                                C00320000
*                   COMPILE=SC02ACAT,                            C00330000
*                   D/SNAME=DMS001A.LMOD,                        C00330000
*                   VERSION=1.0,                                 C00350000
*                   GOODRC=4,                                    C00360000
*                   PORDER=1,                                    C00110000
*                   OPTIONS=(P)                                  00370000
* 1          -- SYSPRINT--                                     00520000
*          FLMALLOC  IOTYPE=W,RECFM=FBA,LRECL=121,               C00530000
*                   RECNUM=20000,PRINT=Y,DDNAME=SYSPRINT         00540000
* 2          -- ERRLIST --                                     00520000
*          FLMALLOC  IOTYPE=W,RECFM=FBA,LRECL=121,               C00530000
*                   RECNUM=20000,PRINT=Y,DDNAME=ERRLIST         00540000
* 3          -- OUT --                                         00520000
*          FLMALLOC  IOTYPE=W,RECFM=FB,LRECL=80,RECNUM=3000,    C00530000
*                   DDNAME=OUT                                   00540000
* 4          -- S2ASW-- --                                     00520000
*          FLMALLOC  IOTYPE=A,DDNAME=S2ASW                       00530000
*                   FLMCPYLB SYS1.CRD
*                   FLMCPYLB SYS1.TESTCRD
*                   FLMCPYLB SYS1.URD1
* 5          -- S2ASW1 --                                       00520000
*          FLMALLOC  IOTYPE=W,RECFM=FB,LRECL=83,RECNUM=2000,    C00530000
*                   DDNAME=S2ASW1,BLKSIZE=1079                  00540000
* 6          -- S2ASW2 --                                       00520000
*          FLMALLOC  IOTYPE=W,RECFM=FB,LRECL=83,RECNUM=2000,    C00530000
*                   DDNAME=S2ASW2,BLKSIZE=1079                  00540000
* 7          -- TEXTLIB --                                       00520000
*          FLMALLOC  IOTYPE=A,DDNAME=TEXTLIB                     00530000
*                   FLMCPYLB NULLFILE
* 8          -- IN --                                           00520000
*          FLMALLOC  IOTYPE=S,KEYREF=SINC,RECFM=FB,LRECL=80,    C00530000
*                   RECNUM=2000,DDNAME=IN                        00540000
*                                                                00230000
*=====
* STEP 2 :  PLI COMPILE                                         00300000
*=====
*                                                                00290000
*          -- PLI COMPILE --                                     00300000
*          FLMTRNSL  CALLNAM='PLI COMPILE',                       C00310000
*                   FUNCTN=BUILD,                                C00320000
*                   COMPILE=IEL1AA,                              C00330000

```

DSNAME=SYS1.IEL.SIELCOMP,	C00330000
VERSION=1.1,	C00350000
GOODRC=8, <-- コンパイラからの戻りコードの許容範囲	C00360000
PORDER=3,	C00110000
PARMKWD=PARM2,	C00110000
OPTIONS=(NIS,A,X,AG,M,MAR(2,72,1)) <-- コンパイラオプション	
* ----- DDNAME SUBSTITUTION LIST -----	
*	
* 1 -- SYSLIN --	00520000
FLMALLOC IOTYPE=0,KEYREF=OBJ,RECFM=FB,LRECL=80,	C
RECNUM=5000,DDNAME=SYSLIN,DFLTYP=OBJ <-- プロジェクトで定義した出力タイプを省略時のタイプとして指定。(この場合はコンパイルされた結果、デフォルトでppppl.xxxLIB.OBJに書き出される)	
* 2 -- N/A --	00520000
FLMALLOC IOTYPE=N	
* 3 -- N/A --	00520000
FLMALLOC IOTYPE=N	
* 4 -- SYSLIB --	00520000
FLMALLOC IOTYPE=I,KEYREF=SINC	
FLMCPYLB SYS1.DCL	
FLMCPYLB SYS1.CTRL2	
FLMCPYLB SYS1.CTRL	
FLMCPYLB SYS1.UCTRL	
FLMCPYLB SYS1.SCEESAMP	
<-- アプリケーション独自のMACROライブラリーまたはINCLUDEがあればここに追加する。	
* 5 -- SYSIN --	00520000
FLMALLOC IOTYPE=A	
FLMCPYLB NULLFILE	
* 6 -- SYSPRINT --	00520000
FLMALLOC IOTYPE=0,RECFM=VBA,LRECL=125,	C
RECNUM=5000,PRINT=Y	
* 7 -- SYSPUNCH --	00520000
FLMALLOC IOTYPE=A,DDNAME=SYSIN	
FLMCPYLB NULLFILE	
* 8 -- SYSUT1 --	00520000
FLMALLOC IOTYPE=W,RECFM=F,LRECL=4051,RECNUM=5000	
* 9 -- N/A --	00520000
FLMALLOC IOTYPE=N	
* 10 -- N/A --	00520000
FLMALLOC IOTYPE=N	
* 11 -- N/A --	00520000
FLMALLOC IOTYPE=N	
* 12 -- N/A --	00520000
FLMALLOC IOTYPE=N	
* 13 -- N/A --	00520000
FLMALLOC IOTYPE=N	
* 14 -- SYSCIN --	00520000
FLMALLOC IOTYPE=U,DDNAME=OUT	
* 15 -- SYSLIB2 --	00520000
FLMALLOC IOTYPE=I,KEYREF=SINC,DDNAME=SYSLIB2	


```
FLMCPYLB SA00.DS.DCL
FLMCPYLB SYS1.CTRL2
FLMCPYLB SYS1.CTRL
FLMCPYLB SYS1.UCTRL
* 16      -- SYSLIB3 --                                00520000
      FLMALLOC IOTYPE=I,KEYREF=SINC,DDNAME=SYSLIB3
      FLMCPYLB SYS1.UCTRL
```

2部 応用編

1. バッチ・ロードモジュール編

2. PSB/DBD編

3. SDTM編

4. アプリケーション編

第1章 バッチ・ロードモジュール編

(TSO/IMSネイティブ/CICSネイティブも同様)

1.1 NON-DB2 W/O NCAL

1.2 NON-DB2 with NCAL

1.3 DB2 W/O NCAL

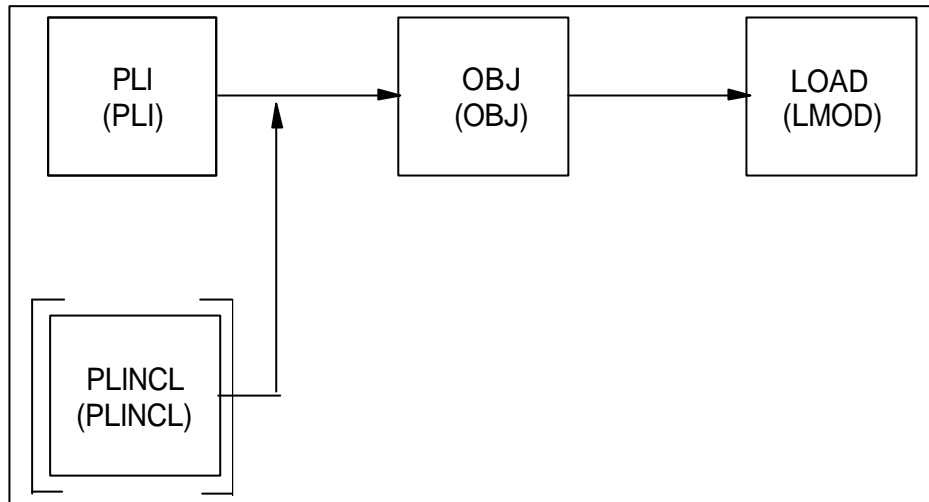
1.4 DB2 with NCAL

1.1 NON-DB2 W/O NCAL

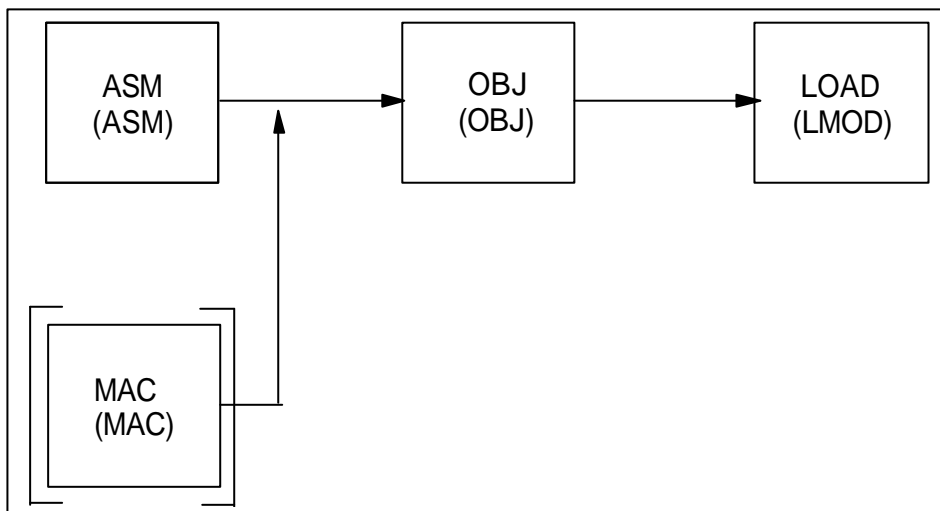
ロードモジュール1個に対し、アーキテクチャー・メンバー1個をARCHDEFに作成する。

メンバー名=ロードモジュール名

例1) PLI



例2) アセンブラ



1.1 NON-DB2 W/O NCAL

<サンプル>

ARCHDEF (memberA)

LKED	LINK2
LOAD	memberA LMOD
INCLD	memberA PLI

ARCHDEF (memberB)

LKED	LINK2
LOAD	memberB LMOD
INCLD	memberB ASM

別名をつける

LKED	LINK2
LOAD	member名 LMOD
INCLD	member名 PLI
ALIAS	別名1 LMOD
ALIAS	別名2 LMOD

INCLUDEする

LKED	LINK2
LOAD	member名 LMOD
INCLD	member名 PLI
CMD	INCLUDE SYSLIB(メンバー名)
CMD	INCLUDE SYSLIB(メンバー名)

リンク・オプションを追加する

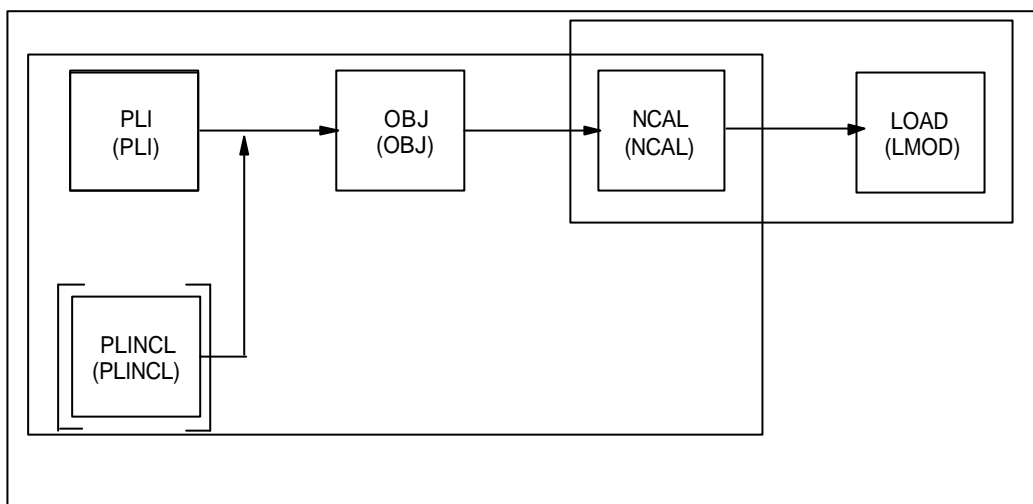
LKED	LINK2
LOAD	member名 LMOD
INCLD	member名 PLI
PARM	MAP , XREF , LET

1.2 NON-DB2 with NCAL

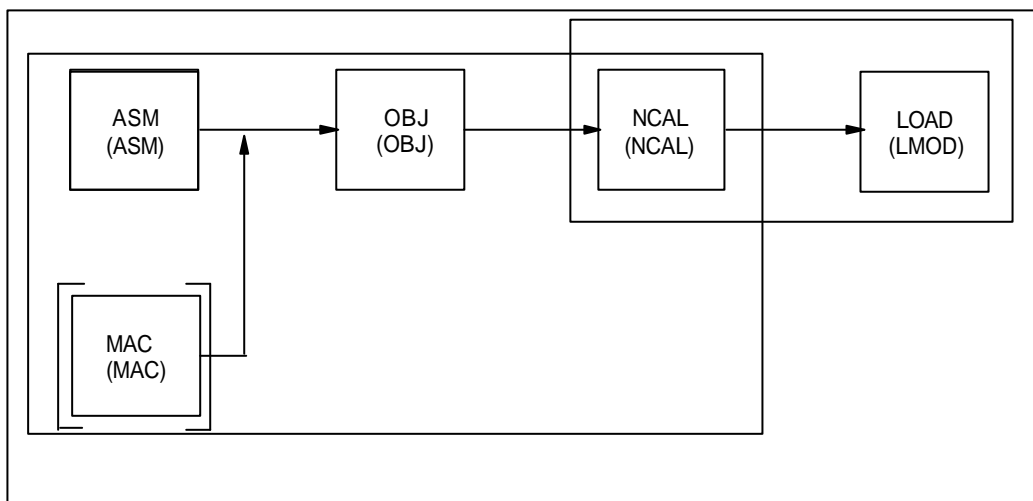
NCAL1個に対し、アーキテクチャー・メンバー1個をARCHNCLに作成する。
メンバー名=NCAL名

ロードモジュール1個に対し、アーキテクチャー・メンバー1個をARCHDEFに作成する
メンバー名=ロードモジュール名

例1) PLI



例2) アセンブラ



1.2 NON-DB2 with NCAL

<サンプル>

ソース NCAL

ARCHNCL (memberA)

LKED	LINKN
LOAD	memberA NCAL
INCLD	memberA PLI

ARCHNCL (memberB)

LKED	LINKN
LOAD	memberB NCAL
INCLD	memberB ASM

NCAL ロード

ARCHNCL (memberA)

LKED	LINK2
LOAD	memberA LMOD
INCL	memberA ARCHNCL

ARCHNCL (memberB)

LKED	LINK2
LOAD	memberB LMOD
INCL	memberB ARCHNCL

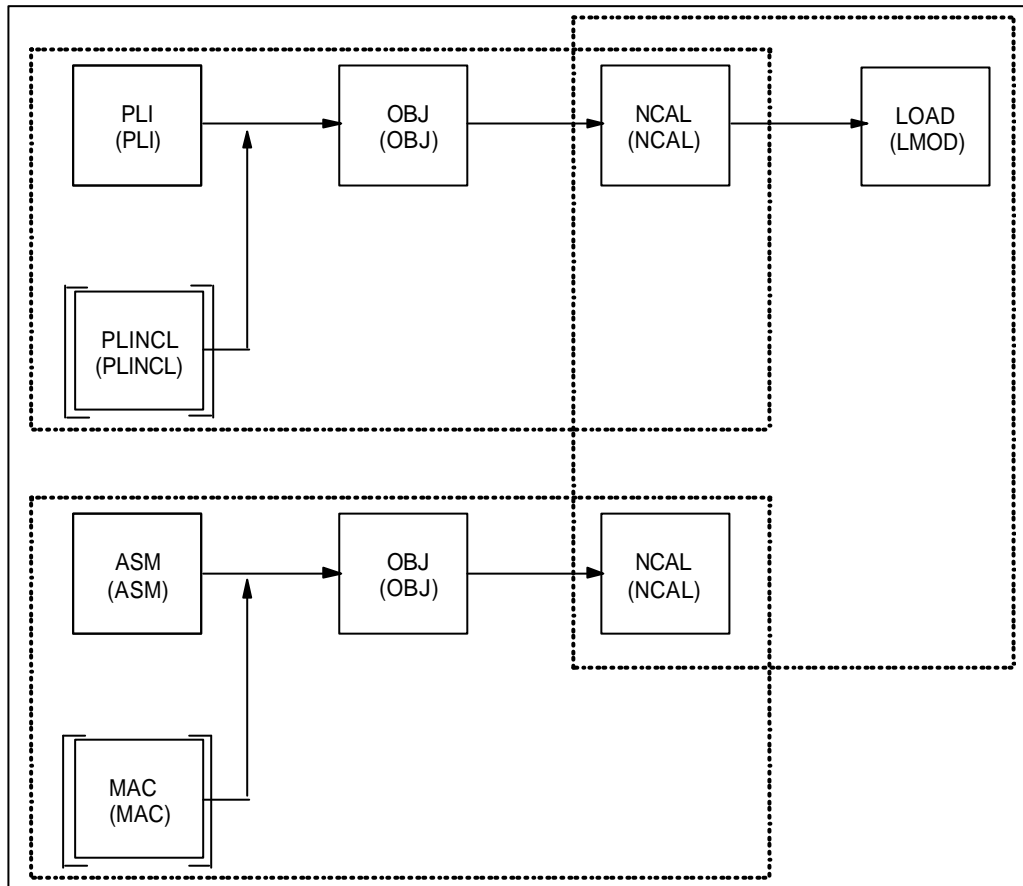
NCAL用アーキテクチャー定義作成ツールあり

「* APPENDIX * 付1」参照

1.2 NON-DB2 with NCAL

応用

例) PLI&アセンブラ



1.2 NON-DB2 with NCAL

<応用のサンプル>

ソース NCAL

ARCHNCL (memberA)

LKED	LINKN	
LOAD	memberA	NCAL
INCLD	memberA	PLI

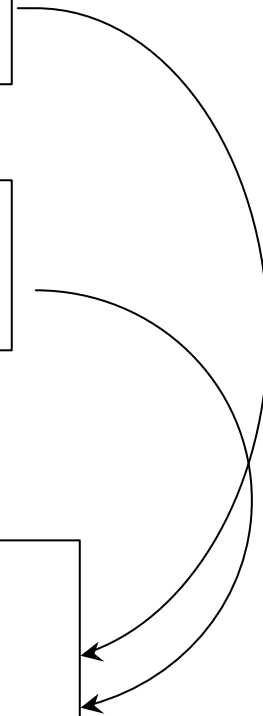
ARCHNCL (memberB)

LKED	LINKN	
LOAD	memberB	NCAL
INCLD	memberB	ASM

NCAL ロード

ARCHDEF (memberB)

LKED	LINK2	
LOAD	memberC	LMOD
INCL	memberA	ARCHNCL
INCL	memberB	ARCHNCL

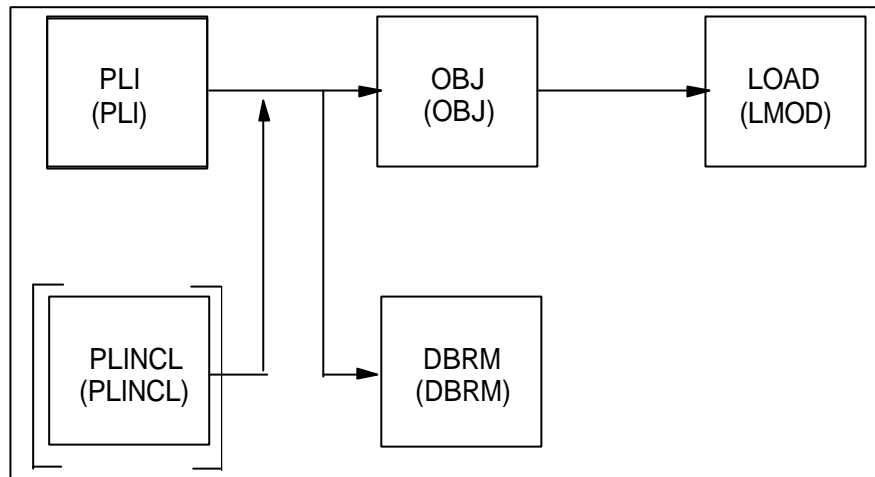


1.3 DB2 W/O NCAL

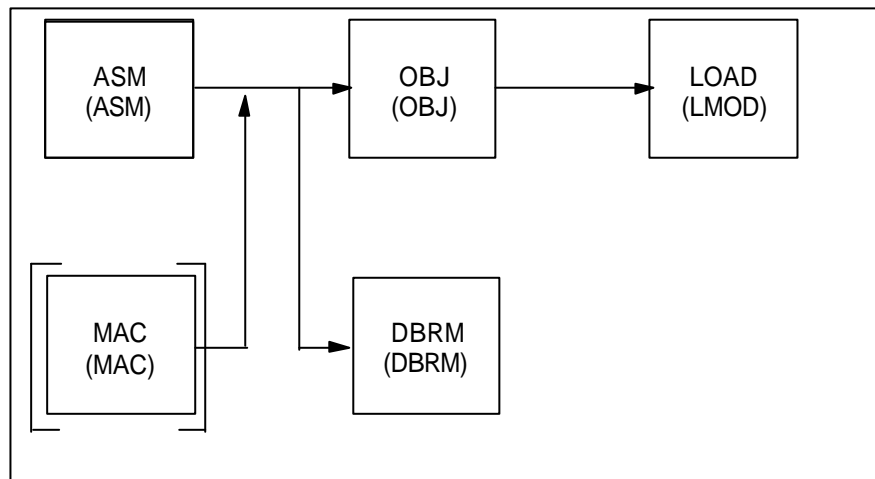
ロードモジュール1個に対し、アーキテクチャー・メンバー1個をARCHDEFに作成する

メンバー名=ロードモジュール名

例1) PLI



例2) アセンブラ



1.3 DB2 W/O NCAL

<サンプル>

ARCHDEF (memberA)

LKED	LINKN	
LOAD	memberA	LMOD
INCLD	memberA	PLI

ARCHDEF (memberB)

LKED	LINKN	
LOAD	memberB	LMOD
INCLD	memberB	ASM

アーキテクチャー・メンバーの内容は、NON-DB2と同じだが、ソース・メンバーの言語は異なる

DB2用の言語定義には、プリコンパイル処理が定義されている。

NON-DB2	PLIソース言語は	「PLI2」
NON-DB2	CICS-PLIソースの言語は	「PLIC」
DB2	PLIソースの言語は	「PLI2DB2」
DB2	CICS-PLIソースの言語は	「PLICDB2」

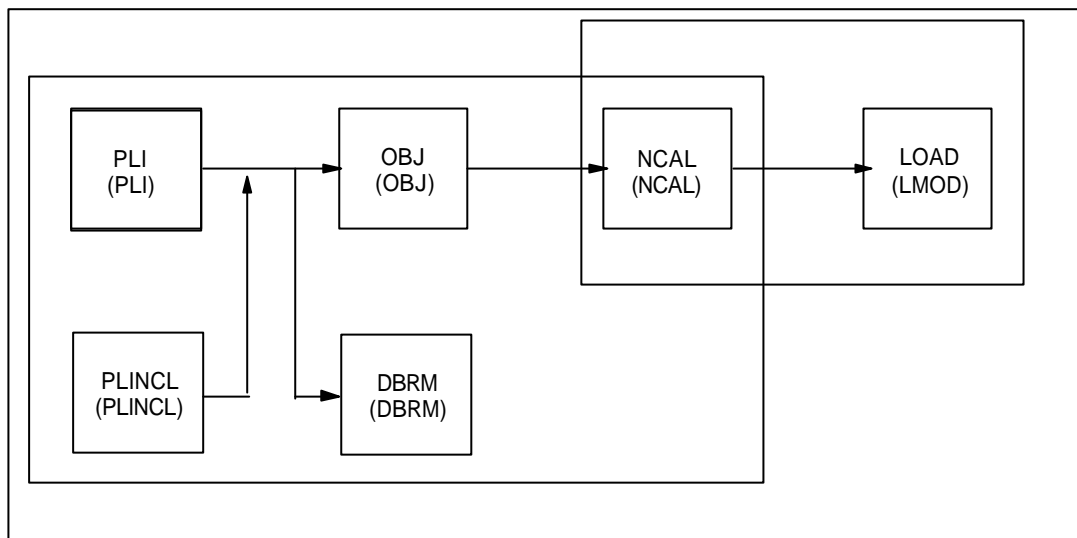
詳しくは「* APPENDIX * 付3」の「AMSサービスで使用する言語定義」を参照

1.4 DB2 with NCAL

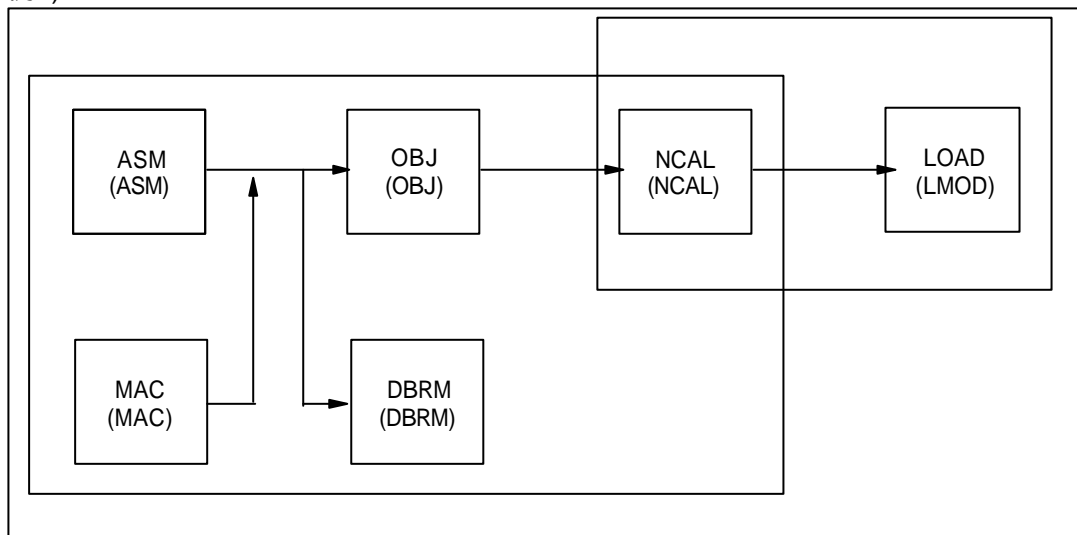
NCAL1個に対し、アーキテクチャー・メンバー1個をARCHNCLに作成する。
メンバー名=NCAL名

ロードモジュール1個に対し、アーキテクチャー・メンバー1個をARCHDEFに作成する
メンバー名=ロードモジュール名

例1) PLI



例2) アセンブラ



1.4 DB2 with NCAL

<サンプル>

ソース NCAL

ARCHNCL (memberA)

LKED	LINKN
LOAD	memberA NCAL
INCLD	memberA PLI

ARCHNCL (memberB)

LKED	LINKN
LOAD	memberB NCAL
INCLD	memberB ASM

NCAL ロード

ARCHDEF (memberA)

LKED	LINK2
LOAD	memberA LMOD
INCL	memberA ARCHNCL

ARCHDEF (memberB)

LKED	LINK2
LOAD	memberB LMOD
INCL	memberB ARCHNCL

アーキテクチャー・メンバーの内容は、NON-DB2と同じだが、ソース・メンバーの言語は異なる

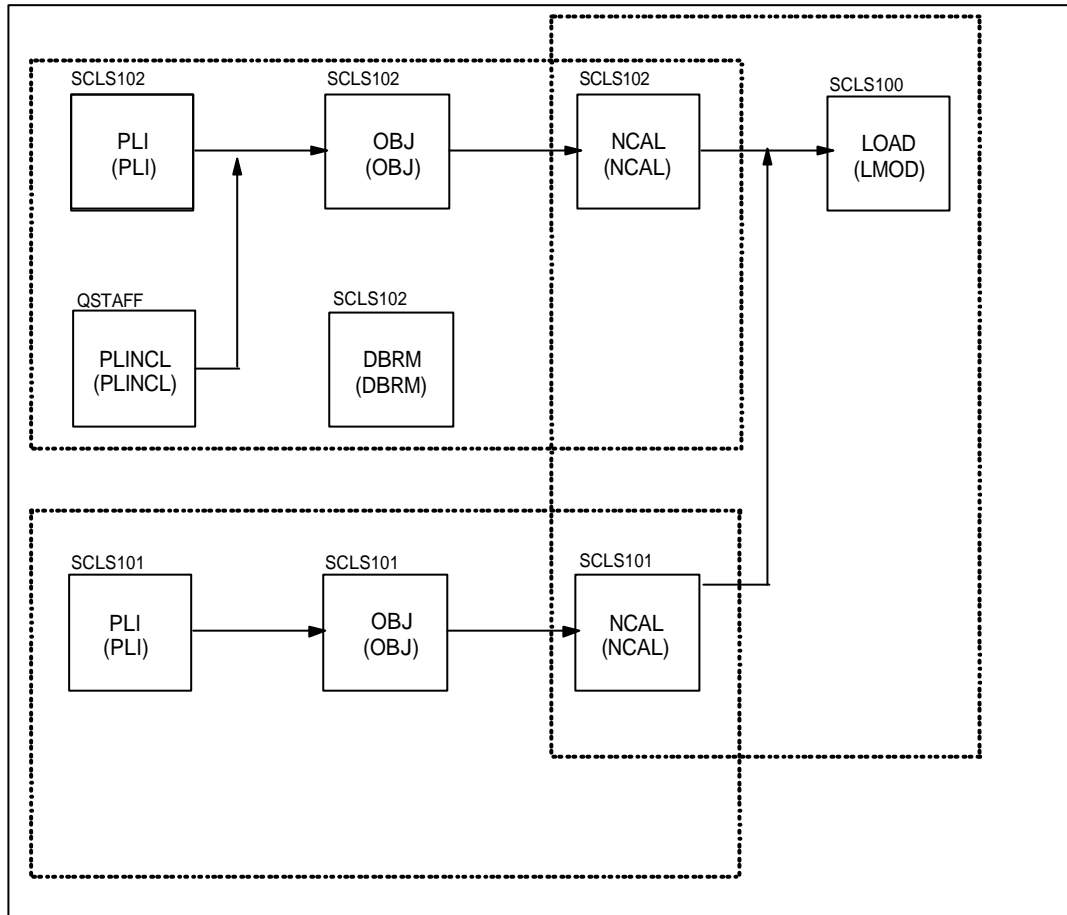
NCAL用アーキテクチャー定義作成ツールあり

「* APPENDIX * 付1」参照

1.4 DB2 with NCAL

応用

例) PLI-DB2



1.4 DB2 with NCAL

<応用のサンプル>

```

編集---TD67.DEVLIB1.ARCHNCL(SCLS101)-01.00----- 桁 001 072
コマンド ==>                                     画面移動 ==> CSR
***** データの始め *****
000100 *
000200 * SCLS101 : LEC ARCHITECTURE DEFINITION
000300 *
000400 LKED LINKN
000500 LOAD SCLS101 NCAL
000600 INCLD SCLS101 PLI
***** データの終わり *****

```

```

編集---TD67.DEVLIB1.ARCHNCL(SCLS102)-01.00----- 桁 001 072
コマンド ==>                                     画面移動 ==> CSR
***** データの始め *****
000100 *
000200 * SCLS102 : LEC ARCHITECTURE DEFINITION
000300 *
000400 LKED LINKN
000500 LOAD SCLS102 NCAL
000600 INCLD SCLS102 PLI
***** データの終わり *****

```

```

編集---TD67.DEVLIB1.ARCHNCL(SCLS101)-01.00----- 桁 001 072
コマンド ==>                                     画面移動 ==> CSR
***** データの始め *****
000001 *
000002 * SCLS100 : LEC ARCHITECTURE DEFINITION
000003 *
000004 LKED LINK2
000005 LOAD SCLS100 LMOD * LOAD MODULE
000006 INCL SCLS101 ARCHNCL * PLI SOURCE1 <
000007 INCL SCLS102 ARCHNCL * PLI SOURCE2 <
***** データの終わり *****

```

第2章 PSB/DBD編

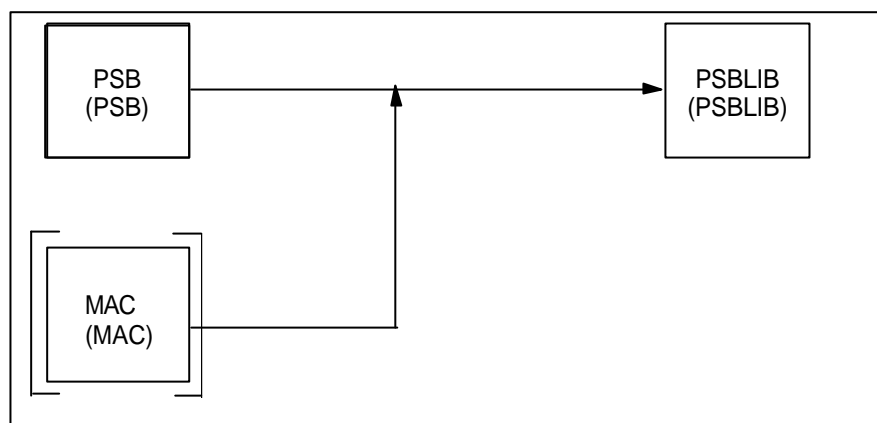
2.1 PSB

2.2 DBD

2.1 PSB

PSBの「組立て」は、ALIAS(別名)を作成する場合と、作成しない場合とで方法が異なる。1アプリケーションで、2つの方法を使用することはできないので、どちらを使用するかは事前に決定すること。大量の別名を作成するアプリケーションを除き、別名を作成しないほうを選択すること。

別名を作成しない場合



アーキテクチャー・メンバーは不要

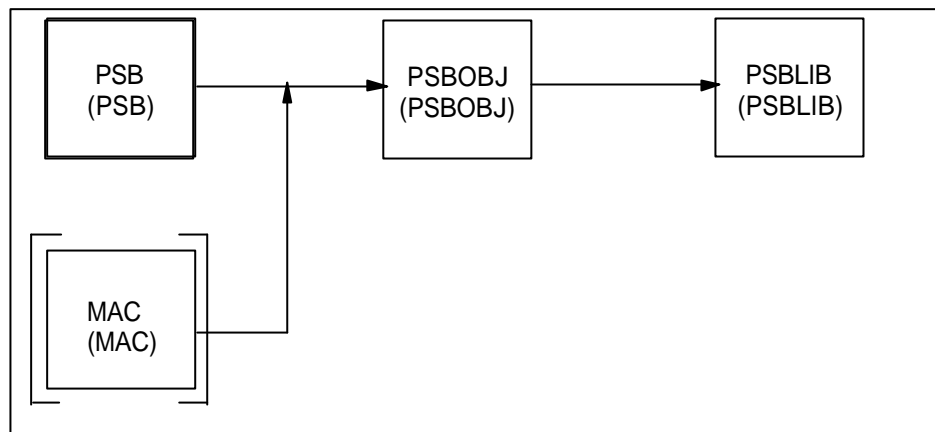
PSBソースの言語定義は「PSB」

「組立て」をする時は、原始ソース・メンバーを指定する

SCLM - 組立て			
コマンド	====>		EXECUTかSUBMIT
<div style="display: flex; justify-content: space-between;"> <div style="width: 60%;"> <p>組立て入力：</p> <p>プロジェクト . . . : TD77</p> <p>グループ DEVLIB1</p> <p>タイプ PSB</p> <p>メンバー JALTP1AC</p> </div> <div style="width: 35%;"> <p>代替 ====>TD77</p> <p>← 原始ソース・メンバー</p> </div> </div>			

2.1 PSB

別名を作成する場合



ロードモジュール1個に対し、アーキテクチャー・メンバー1個をARCHDEFに作成する

メンバー名=ロードモジュール名

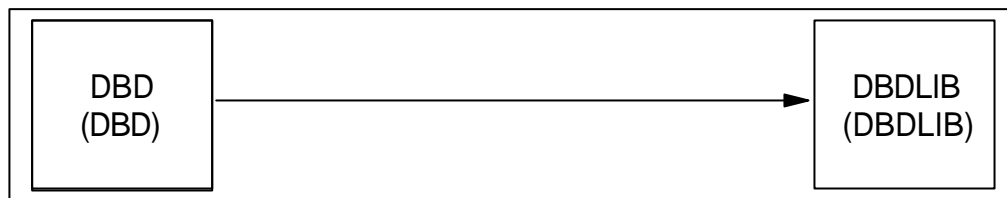
PSBソースの言語定義は「PSBOBJ」

< サンプル >

ARCHDEF (memberA)

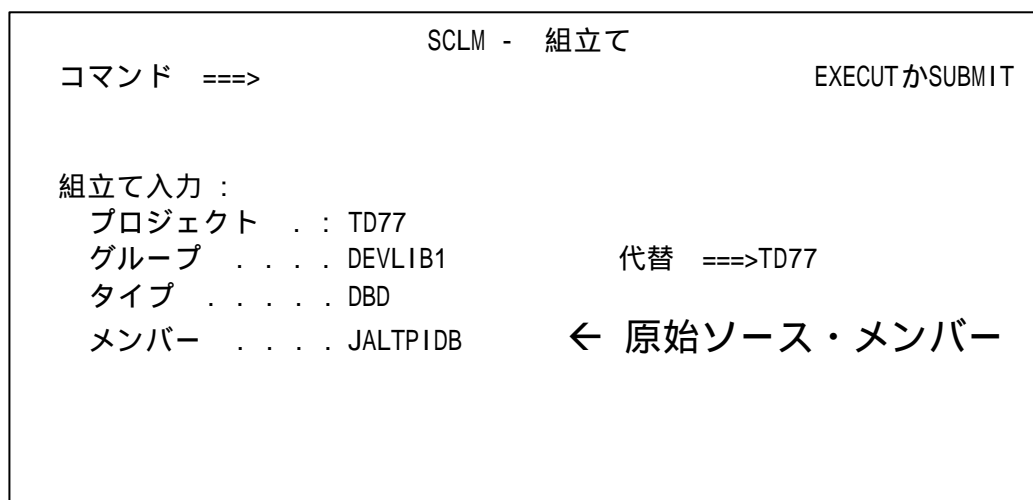
LKED	PSBLNK	
LOAD	memberA	PSBLIB
INCLD	memberA	PSB
ALIAS	別名1	PSBLIB
ALIAS	別名2	PSBLIB

2.2 DBD



アーキテクチャー・メンバーは作成しない

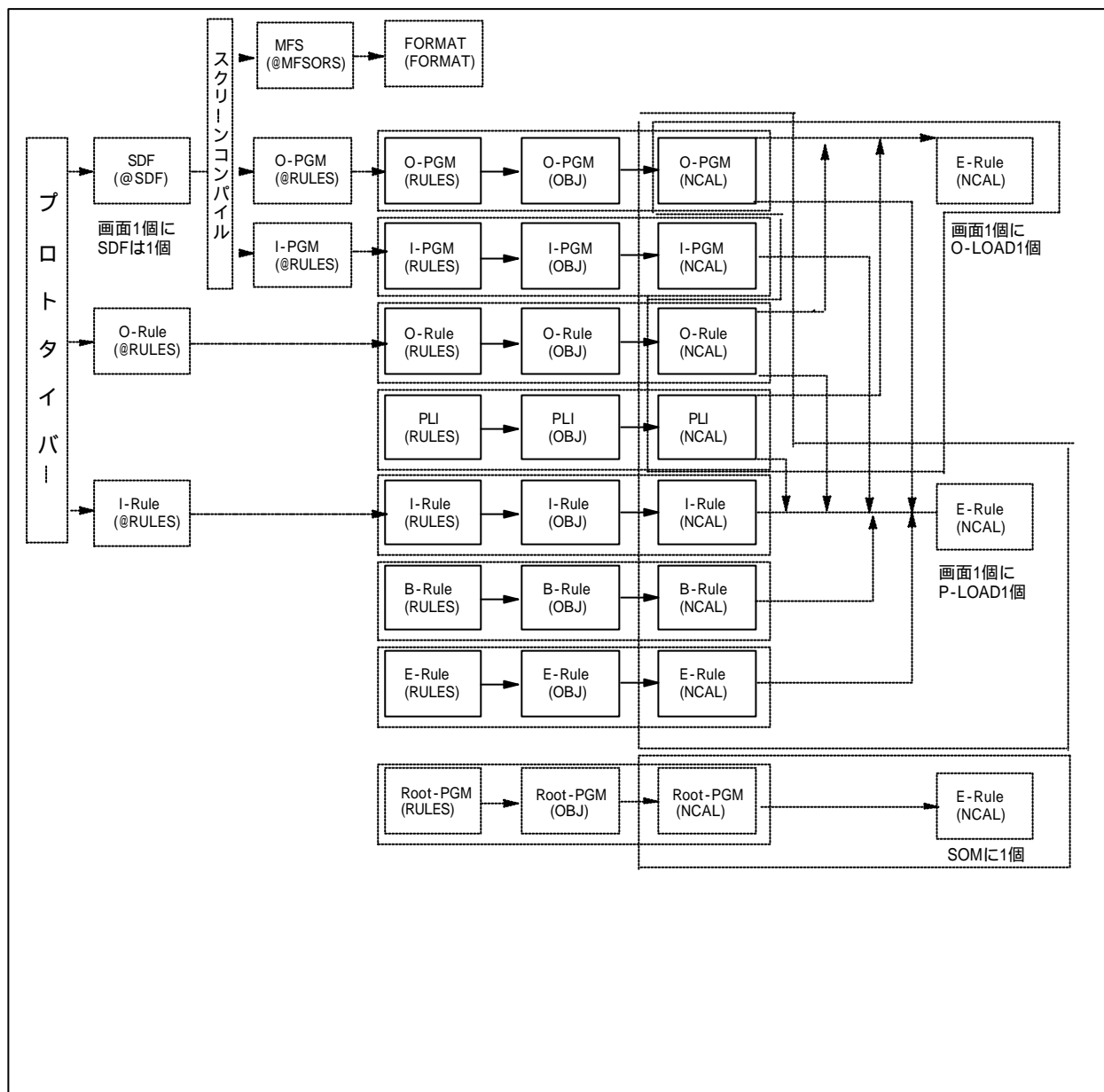
「組立て」をする時は、原始ソース・メンバーを指定する



第3章 SDTM編

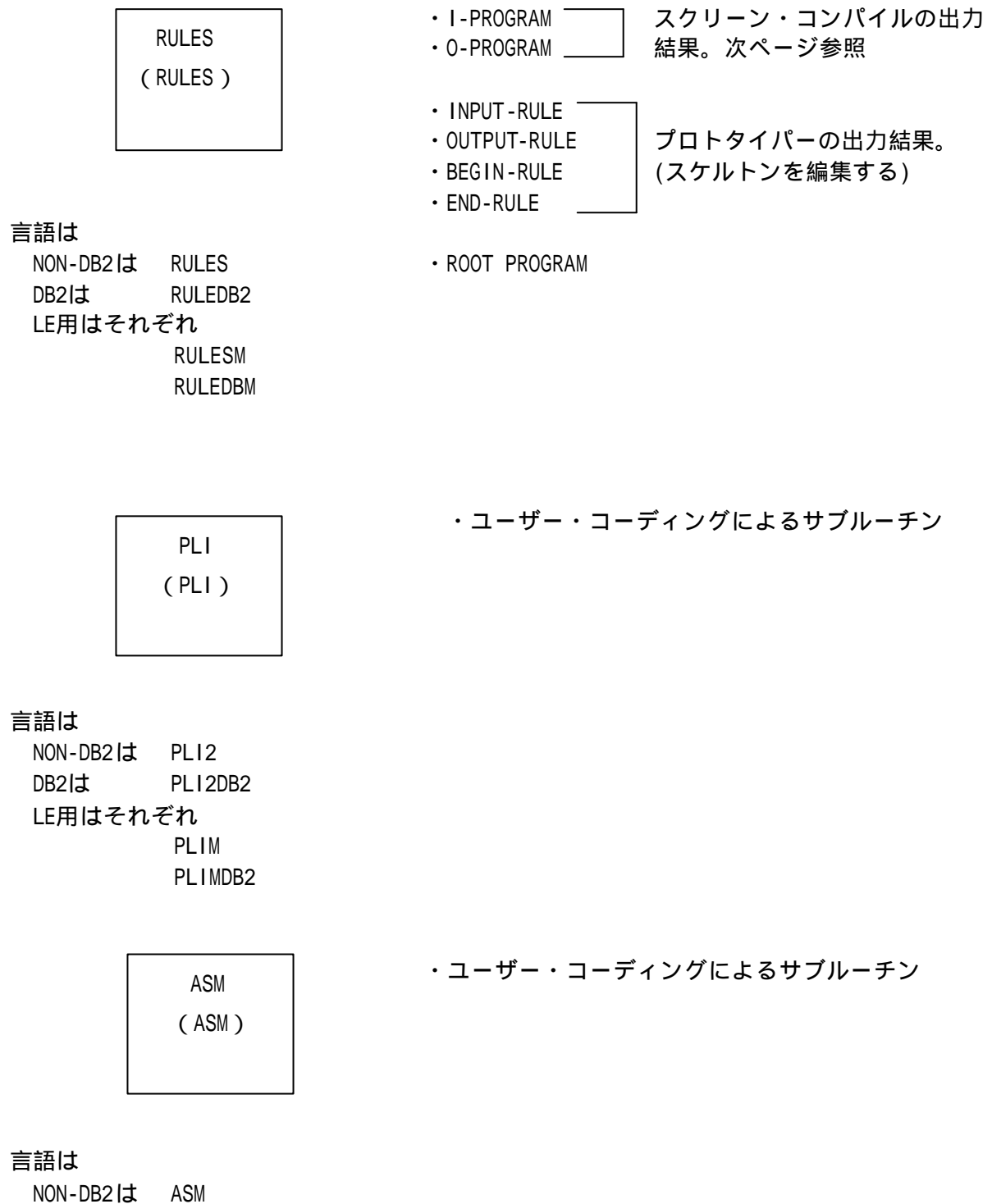
- 3.1 概要
- 3.2 NCALモジュール
- 3.3 O-LOADモジュール
- 3.4 P-LOADモジュール
- 3.5 ROOT-LOADモジュール

3.1 概要



3.1 概要

ソースの種類



3.1 概要

I/O PROGRAMのソースの取得方法

スクリーン・コンパイルを実施する際に、“ICODE”と“OCODE”のDDカードを追加する。

<サンプルのスクリーン・コンパイル用JCL>

```
//      E08641CP  JOB  (F9500B,SQ23X,32),E08641KUWANO,CLASS=M,          00001000
//      MSGCLASS=R TYPRUN=SCAN                                          00002000
//*****                                                                00003000
//* DESCRIPTION - PROCEDURE TO EXECUTE SCREEN COMPILER TO PRODUCE * 00004000
//*           I AND O PROGRAMS AND MFS BLOCKS FROM SCREEN DEFINITIONS * 00005000
//*                                                                * 00006000
//*                                                                * 00007000
//* NOTE :                                                                * 00008000
//*      SDLIB   = SCREEN DEFINITION SOURCE LIBRARY                    * 00009000
//*      MBR     = SCREEN DEFINITION SOURCE LIBRARY MEMBER NAME        * 00010000
//*      SCPARM  = SCREEN COMPILER PARAMETER OPTIONS                   * 00020000
//*      STRCLIB = USERS STRUCTURE LIBRARY FOR %INCLUDES                * 00030000
//*      INCLIB  = USERS ALTERNATE LIBRARY FOR %INCLUDES                * 00040000
//*                                                                * 00050000
//* A DATASET "&APPL3..&LVL3..MFSORS" IS GENERATED WHICH CONTAINS * 00060000
//* THE NECESSARY MFS SOURCE STATEMENTS AND MUST BE PASSED            * 00070000
//* THROUGH THE MFS GEN PROCEDURE "RWBCMFS" OR EQUIVALENT.            * 00080000
//*                                                                * 00090000
//* THE SCPARM PARAMETER REFERS TO A MEMBER IN THE DATA LIBRARY      * 00100000
//* THE MEMBER NAME INDICATES THE RESULT OF ITS USE AS A              * 00110000
//* PARAMETER TO THE SCREEN COMPILER.                                  * 00120000
//*   THE FIRST 2 BYTES - M# - ARE INSTRUCTIONS CONCERNING MFS.        * 00130000
//*   THE SECOND 2 BYTES - I# - ARE INSTRUCTIONS CONCERNING I-PGM.    * 00140000
//*   THE THIRD 2 BYTES - O# - ARE INSTRUCTIONS CONCERNING O-PGM.     * 00150000
//* THE # IN THE NAME IS REPLACEDBY ONE OF THE FOLLOWING :-          * 00160000
//*       G - EDIT AND GENERATE                                         * 00170000
//*       E - EDIT ONLY.                                                * 00180000
//*       N - NO PROCESSING REQUIRED.                                     * 00190000
//*       F - EDIT AND GENERATE - O-PGM LOADFORMAT. (I-PGM ONLY)* 00200000
//*                                                                * 00210000
//*****                                                                00220000
//* RERUNNABLE      Y                                                  * 00230000
//* STEPRESTART     N                                                  * 00240000
//* STEP010         - NO BACKOUT IS NECESSARY - ONLY RESUBMIT JOB     * 00250000
//*                                                                * 00260000
//*****                                                                00270000
//RWBCPLS  PROC APPL1='ECS.COMM',      * 1ST LEVEL QUALIFIER FOR DSNAME 00280000
//          APPL2=SQ23IVT,              * 1ST LEVEL QUAL. FOR DB DSNAME 00290000
//          APPL3=TD77I,                 * 1ST LEVEL QUAL. FOR APPLN DSN1 00300000
```

```

//      IMSRLIB='IMSVS.RESLIB', * IMS/VS RESIDENT LIBRARY      00310000
//      INCLIB='SYS1.DCL', * 2ND INCLUDE LIBRARY      00320000
//      LVL1=SYSTEM, * 2ND LEVEL QUALIFIER FOR DSNAM      00330000
//      LVL2=BDLI, * 2ND LEVEL QUAL. FOR DB DSNAM      00340000
//      LVL3=TDS, * 2ND LEVEL QUAL. FOR APPLN DSN      00350000
//      LVL4=SYSTEM, 00360000
//      MBRB01=DUMMY, * SDTM SCREEN NAME      00370000
//      OUTD='*', * DUMP OUTPUT CLASS      00380000
//      OUTP='*', * OUTPUT CLASS (SYSPRINT)      00390000
/      REGION3=2048K, * REGION SIZE 3      00400000
//      SCPARM=MGIGOG, * SDTM COMPILER OPTIONS      00410000
// *      SPACB01='(CYL,(2,1),RLSE,,ROUND)', * SDTM SPACE 1      00420000
// *      STRCLIB=, * 1ST INCLUDE LIBRARY      00430000
// *      SYSDA=SYSDA, * TEMPORARY FILE DEVICE      00440000
// *** IMS/VS EXECUTION PARAMETERS      00450000
//      BKO=, * DYNAMIC BACKOUT      00460000
//      BUF=20, * BUFFER POOL SIZE      00470000
//      CKPTID=, * CHECKPOINT IDENTIFIER      00480000
//      DBRC=, * DB RECOVERY CONTROL OPTION      00490000
//      EXCPVR=0, * PAGEFIX BUFFER POOL OPTION      00500000
//      FMTO=T, * FORMATTED DUMP OPTION      00510000
//      IMSID=, * IMS IDENTITY OVERRIDE      00520000
//      IRLM=, * IRLM OPTION      00530000
//      IRLNM=, * IRLM SUB-SYSTEM NAME      00540000
//      LOGA=0, * BSAM/OSAM LOGGING ACCESS      00550000
//      MON=N, * DB MONITOR ACTIVE OPTION      00560000
//      PRLD=, * PRELOAD SUFFIX MEMBER      00570000
//      RST=0, * UCF RESTART OPTION      00580000
//      SPIE=0, * SPIE OPTION      00590000
//      SRCH=0, * MODULE SEARCH INDICATOR      00600000
//      SWAP=, * ADDRESS SPACE SWAPPABLE      00610000
//      TEST=0 * TEST OPTION      00620000
//      00630000
//----- 00640000
// STEP010 SDTM SCREEN COMPILER      00650000
//----- 00660000
//      00670000
// STEP010 EXEC PGM=DFSRRCOO,      00680000
//      REGION=&REGION3,      00690000
//      PARM=(DLI,RWBCOOP,RWBCOOP,&BUF,      00700000
//      &SPIE&TEST&EXCPVR&RST,&PRLD,      00710000
//      &SRCH,&CKPTID,&MON,&LOGA,&FMTO,      00720000
//      &IMSID,&SWAP,&DBRC,&IRLM,&IRLMNM,&BKO)      00730000
//STEPLIB DD DSN=&IMSRLIB,DISP=SHR      00740000
//      DD DSN=&APPL1..&LVL4..EXECLOAD,DISP=SHR      00750000
//      DD DSN=SYS1.PLIBASE,DISP=SHR      00760000
//      DD DSN=SYS1.PLI.PLICOMP,DISP=SHR      00770000
//      DD DSN=SYS1.PLI.PLILINK,DISP=SHR      00780000

```


//	DD DSN=SYS1.PLITASK,DISP=SHR	00790000
//	DD DSN=SYS1.SIBMBASE,DISP=SHR	00800000
//	DD DSN=SYS1.PLI.SIBMLINK,DISP=SHR	00810000
//DFSRESLB	DD DSN=&IMSRLIB,DISP=SHR	00820000
//IMS	DD DSN=IMSVS.DBDLIB,DISP=SHR	00830000
//	DD DSN=IMSVS.PSBLIB,DISP=SHR	00840000
//IEFRDER	DD DUMMY	00850000
//DFSVSAMP	DD DSN=&APPL1..&LVL1..DATA(ECSPools),DISP=SHR	00860000
//PRINTER	DD SYSOUT=&OUTP,	00870000
//	DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)	00880000
//SUMMARY	DD SYSOUT=&OUTP,	00890000
//	DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)	00900000
//SYSPRINT	DD SYSOUT=&OUTP	00910000
//PLIPRINT	DD SYSOUT=&OUTP,	00920000
//	DCB=(RECFM=VBA,LRECL=125,BLKSIZE=1573)	00930000
//LYSPRINT	DD SYSOUT=&OUTP,	00940000
//	DCB=BLKSIZE=1210	00950000
//SYSOUT	DD SYSOUT=&OUTP	00960000
//SYSUDUMP	DD SYSOUT=&OUTD	00970000
//PLIDUMP	DD SYSOUT=&OUTP	00980000
//RWBRTAD	DD DSN=&APPL2..&LVL2..RWBRTAD,DISP=SHR	00990000
//RWBRDED	DD DSN=&APPL2..&LVL2..RWBRDED,DISP=SHR	01000000
//SORTLIB	DD DSN=SYS1.SORTLIB,DISP=SHR	01010000
//CSYSLIB	DD DSN=&INCLIB,DISP=SHR	01020000
//	DD DSN=&APPL1..&LVL1..ECSINCL,DISP=SHR	01030000
//	DD DSN=&APPL1..&LVL1..PLI,DISP=SHR	01040000
//	DD DSN=&APPL1..&LVL1..PLINCL,DISP=SHR	01050000
//LSYSLIB	DD DSN=PLI.V2R2M1.PLIBASE,DISP=SHR	01060000
//OSYSLMOD	DD DSN=&APPL3..&LVL3..@LOAD(&MBRB01.0),DISP=OLD	01070000
//*SYSLMOD	DD DSN=&&OSYSLMOD(&MBRB01.0),DISP=(,DELETE,DELETE),	01070100
//*	SPACE=(CYL,(1,1,5)),UNIT=SYSDA,	01070200
//*	DCB=(RECFM=U,LRECL=0,BLKSIZE=13030)	01070300
//ISYSLMOD	DD DSN=&APPL3..&LVL3..@LOAD(&MBRB01.1),DISP=OLD	01070400
//*SYSLMOD	DD DSN=&&ISYSLMOD(&MBRB01.1),DISP=(,DELETE,DELETE),	01070500
//*	SPACE=(CYL,(1,1,5)),UNIT=SYSDA,	01070600
//*	DCB=(RECFM=U,LRECL=0,BLKSIZE=13030)	01070700
//SCPARM	DD DSN=&APPL1..&LVL1..DATA(&SCPARM),DISP=SHR	01070800
//SCRDESC	DD DSN=&APPL3..&LVL3..@SDF(&MBRB01),DISP=SHR	01070900
//MFSORS	DD DSN=&APPL3..&LVL3..@MFSORS(&MBRB01),DISP=OLD	01071000
//OCODE	DD DSN=&APPL3..&LVL3..@RULES(&MBRB01.0),DISP=SHR	01072000
//ICODE	DD DSN=&APPL3..&LVL3..@RULES(&MBRB01.1),DISP=SHR	01073000
//	PEND	01074000
//*	-----	01075000
//STEPS00	EXEC RWBCPLS,MBRB01=JLTD00	01076000
//*	-----	01077000
//STEPS10	EXEC RWBCPLS,MBRB01=JLTD10	01078000
//*	-----	01079000

//STEPS20 EXEC RWBCPLS,MBRB01=JLTD20	01080000
//*-----	01090000
//STEPS30 EXEC RWBCPLS,MBRB01=JLTD30	01100000
//*-----	01110000
//STEPS40 EXEC RWBCPLS,MBRB01=JLTD40	01120000
//*-----	01130000
//STEPS50 EXEC RWBCPLS,MBRB01=JLTD50	01140000
//*-----	01150000
//STEPS51 EXEC RWBCPLS,MBRB01=JLTD51	01160000
//*-----	01170000
//STEPS52 EXEC RWBCPLS,MBRB01=JLTD52	01180000
//*-----	01190000

3.2 NCALモジュール

SDTMにおいて作成される

O-LOADモジュール
P-LOADモジュール
ROOT-LOADモジュール

のすべてにおいて

NCAL1個に対し、アーキテクチャー・メンバー1個をARCHNCLに作成する

メンバー名=NCAL名

ロードモジュール1個に対して、アーキテクチャー・メンバー1個をARCHDEFに作成する

メンバー名=ロードモジュール名

3.2 NCALモジュール

< NCAL用サンプル >

1) RULESの場合

ARCHNCL (memberA)

LKED	LINKN	
LOAD	memberA	NCAL
INCLD	memberA	RULES

NON-DB2 RULESソースの言語は 「RULES」, LE用は 「RULESM」
DB2 RULESソースの言語は 「RULEDB2」, LE用は 「RULEDBM」

2) PLIの場合

ARCHNCL (memberA)

LKED	LINKN	
LOAD	memberA	NCAL
INCLD	memberA	PLI

NON-DB2 PLIソースの言語は 「PLI2」, LE用は 「PLIM」
DB2 PLIソースの言語は 「PLIDB2」, LE用は 「PLIMDB2」

3) ASMの場合

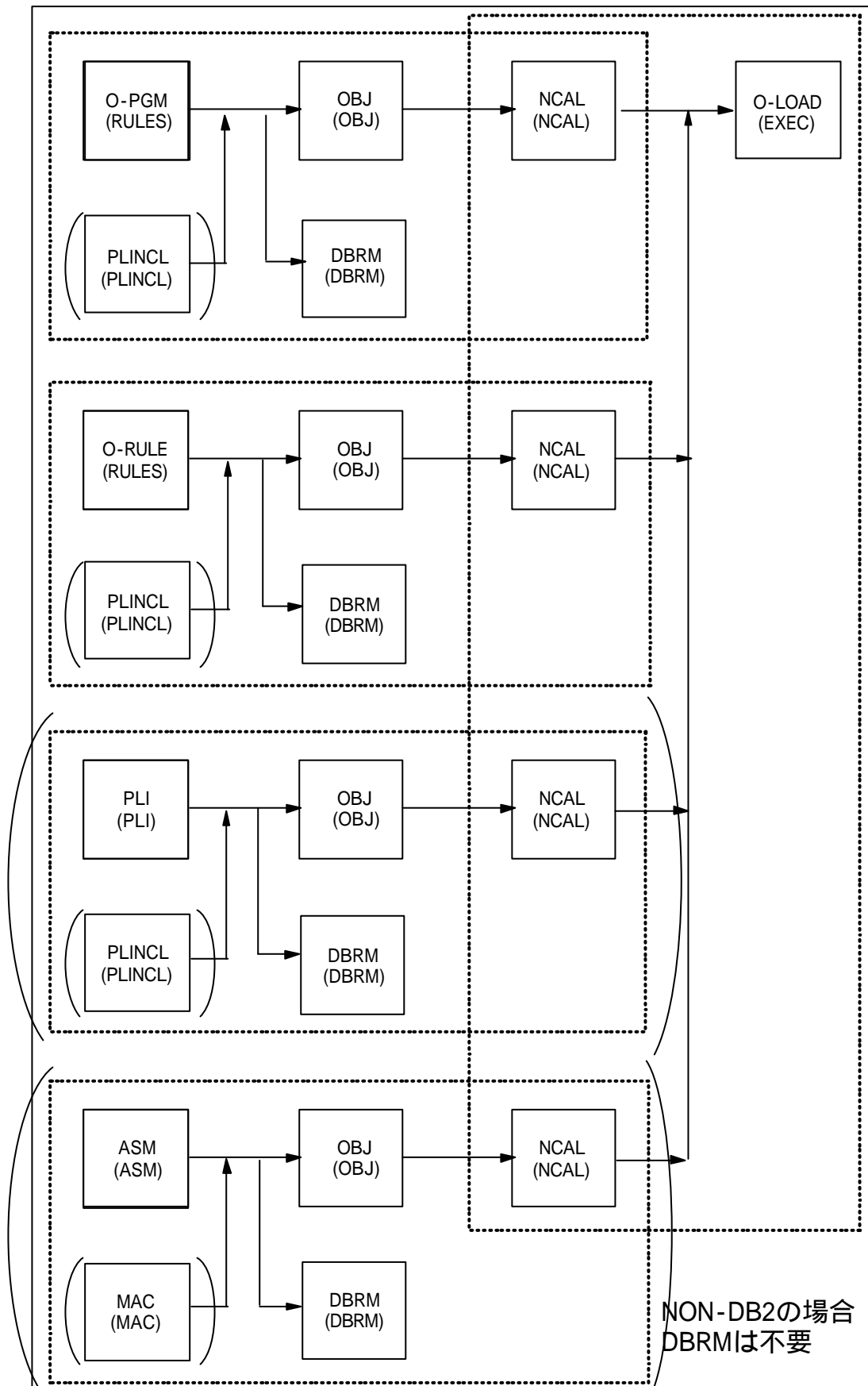
LKED	LINKN	
LOAD	memberB	NCAL
INCLD	memberB	ASM

NON-DB2 ASMソースの言語は 「ASM」

NCAL用アーキテクチャー定義作成ツールあり

「 * APPENDIX * 付1 」 参照

3.3 O-LOADモジュール



3.3 0-LOADモジュール

<0-LOAD用のサンプル>

0-LOADのロードモジュール名は、screenID + 0

ARCHDEF(ロードモジュール名)

(1) LKED	SDTMLNK	
(2) LOAD	ロードモジュール名	EXEC
(3) INCL	0-PGM名	ARCHNCL
(4) INCL	0-RULE名	ARCHNCL
(5) INCL	PLIサブルーチン名	ARCHNCL
(6) INCL	ASMサブルーチン名	ARCHNCL
(7) CMD	INCLUDE	SYSLIB (DFSLI000)
(8) CMD	INCLUDE	SYSLIB (RWBRVUM)
(9) CMD	ENTRY	0-PGM名
(10) CMD	NAME	ロードモジュール名 (R)

- (1) リンクの言語定義はSDTMLNKである。
- (2) 0-LOADのロードモジュール名は、screenID+0であり、ライブラリー・タイプはEXECである。
- (3) 自画面の0-PGMのNCALをリンクする。0-PGM名は、全部リンクする。
- (4) 自画面のOUTPUT-RULEのNCALをリンクする。複数ある場合は、全部リンクする。
- (5) PLIサブルーチンがある場合は、そのNCALをリンクする。
- (6) ASMサブルーチンがある場合は、そのNCALをリンクする。
- (7) DB2の場合、SYSLIBからDFSLI000をINCLUDEする。
- (8) SYSLIBからRWBRVUMをINCLUDEする。
- (9) ENTRYには0-PGM名 (screenID+0) を指定する。
- (10) ロードモジュール名と指定する。

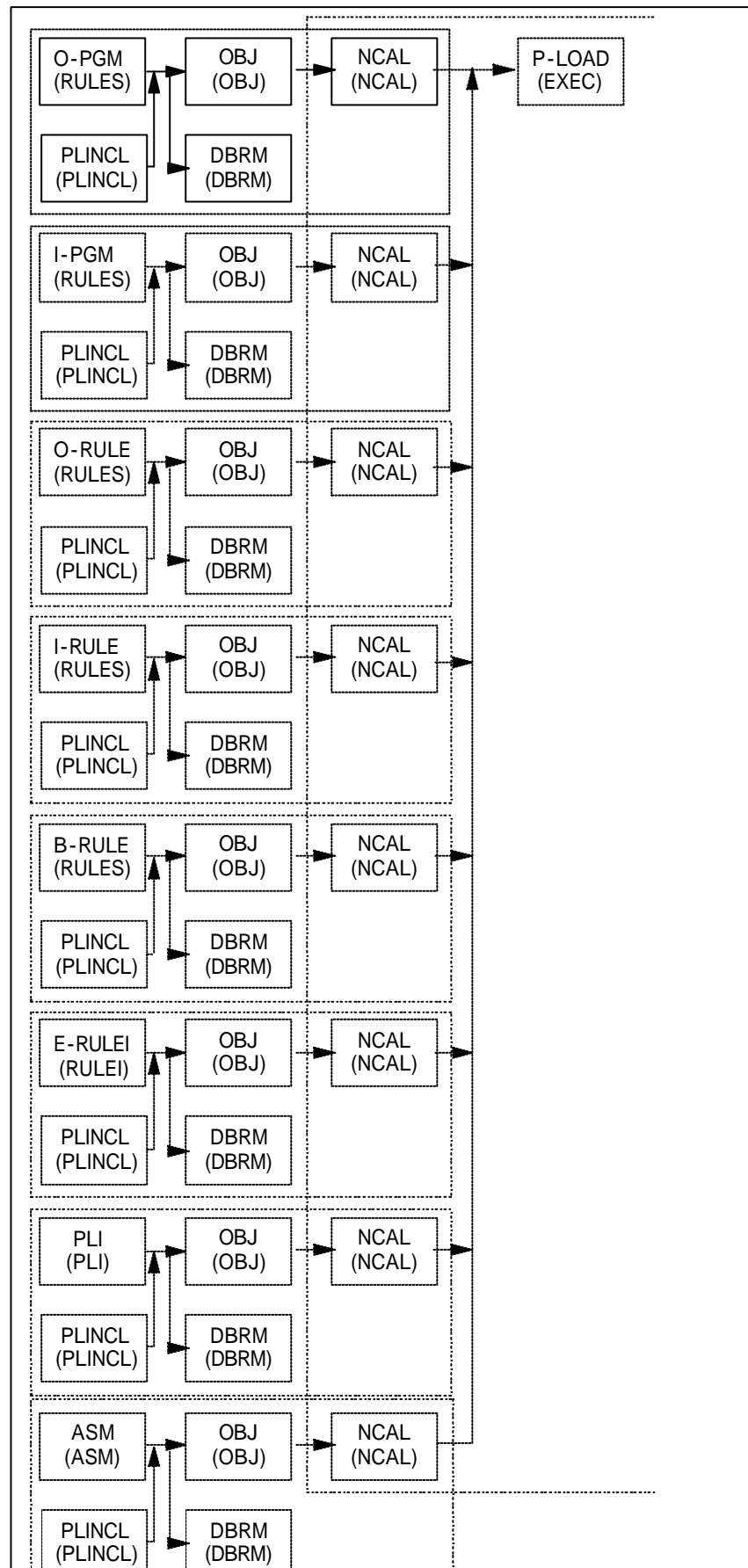
注意事項

0-LOADのリンクのリターンコードは0が正常である。

0/P-LOAD用アーキテクチャー定義作成ツールあり。

「* APPENDIX * 付1」参照

3.4 P-LOADモジュール



3.4 P-LOADモジュール

< P-LOAD用のサンプル >

P-LOADのロードモジュール名は、screenID + P

ARCHDEF(ロードモジュール名)

(1)	LKED	SDTMLNK	
(2)	LOAD	ロードモジュール名	EXEC
(3)	INCL	I-PGM名	ARCHNCL
(4)	INCL	O-RULE名	ARCHNCL
(5)	INCL	I -RULE名	ARCHNCL
(6)	INCL	O-RULE名	ARCHNCL
(7)	INCL	B-RULE名	ARCHNCL
(8)	INCL	E-RULE名	ARCHNCL
(9)	INCL	飛び先画面のO-PGM名	ARCHNCL
(10)	INCL	飛び先画面のO-RULE名	ARCHNCL
(11)	INCL	PLIサブルーチン名	ARCHNCL
(12)	INCL	ASMサブルーチン名	ARCHNCL
(13)	CMD	INCLUDE	SYSLIB (DFSLI000)
(14)	CMD	INCLUDE	SYSLIB (RWBRVUM)
(15)	CMD	ENTRY	O-PGM名
(16)	CMD	NAME	ロードモジュール名 (R)

- (1) リンクの言語定義は SDTMLNK である。
- (2) P-LOADのロードモジュール名は、screenID + Pであり、ライブラリー・タイプはEXECである。
- (3) 自画面のI-PGMのNCALをリンクする。
I-PGM名はscreenID + Iである。
- (4) 自画面のO-PGMのNCALをリンクする。
O-PGM名はscreenID+0である。
- (5) 自画面のINPUT-RULEのNCALをリンクする。
複数ある場合は、全部リンクする。
- (6) 自画面のOUTPUT-RULEのNCALをリンクする。
複数ある場合は、全部リンクする。
- (7) 自画面のBEGIN-RULEのNCALをリンクする。
- (8) 自画面のEND-RULEのNCALをリンクする
- (9) 飛び先画面のO-PGMのNCALをリンクする。
O-PGM名は、飛び先の screenID + 0である。

3.4 P-LOADモジュール

- (10) 先画面のOUTPUT-RULEのNCALをリンクする。
複数ある場合は、全部リンクする。
- (11) PLIサブルーチンがある場合は、そのNCALをリンクする。
- (12) ASMサブルーチンがある場合は、そのNCALをリンクする。
- (13) DB2の場合、SYSLIBからDFSLI000をINCLUDEする。
- (14) SYSLIBから RWBRVUM をINCLUDEする。
- (15) ENTRYには I-PGM名 (screenID+ I)を指定する。
- (16) ロードモジュール名と指定する。

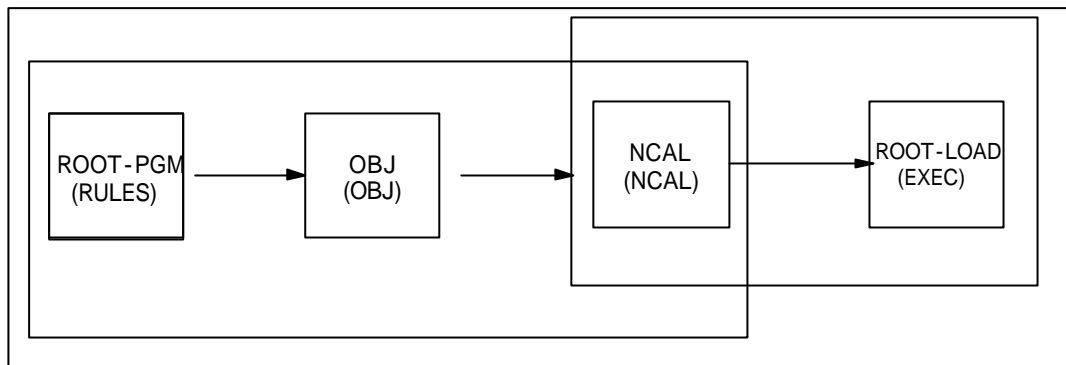
注意事項

P-LOADのリンクのリターンコードは0が正常である。

O/P-LOAD用アーキテクチャー定義作成ツールあり。

「* APPENDIX * 付1」参照

3.5 ROOT-LOADモジュール



< ROOT-LOAD用のサンプル >

ARCHDEF(ロードモジュール名)

(1)	LKED	SDTMLNK	
(2)	LOAD	ロードモジュール名	EXEC
(3)	INCL	Root PGMソース名	ARCHNCL
(4)	CMD	INCLUDE	SYSLIB (RWBROSM)
(5)	CMD	LIBRARY	* (RWBRAOM)
(6)	CMD	ENTRY	RWBROSM
(7)	CMD	NAME	ロードモジュール名 (R)

- (1) リンクの言語定義はSDTMLNKである。
- (2) ライブラリー・タイプはEXECである。
- (3) ROOT-PGMのNCALをリンクする。
- (4) SYSLIBから RWBROSM をINCLUDEする。(non-COMSECはBTSROSMとする)
- (5) このように指定する。
- (6) ENTRYには RWBROSM を指定する。(non-COMSECはBTSROSMとする)
- (7) ロードモジュール名を指定する。

注意事項

ROOT-LOADのリンクのリターンコードは4が正常である。

次のWarningが出る。

IEW0461 RWBRAOM

IEW0461 WARING – SYMBOL PRINTER IS AN UNRESOLVED EXTERNAL REFERENCE;

第4章 アプリケーション編

4.1 概要

4.2 ロードモジュール

4.3 PSB/DBD

4.1 概要

1つのアーキテクチャー・メンバーで複数メンバーの「組立て」や「プロモート」をおこなう。

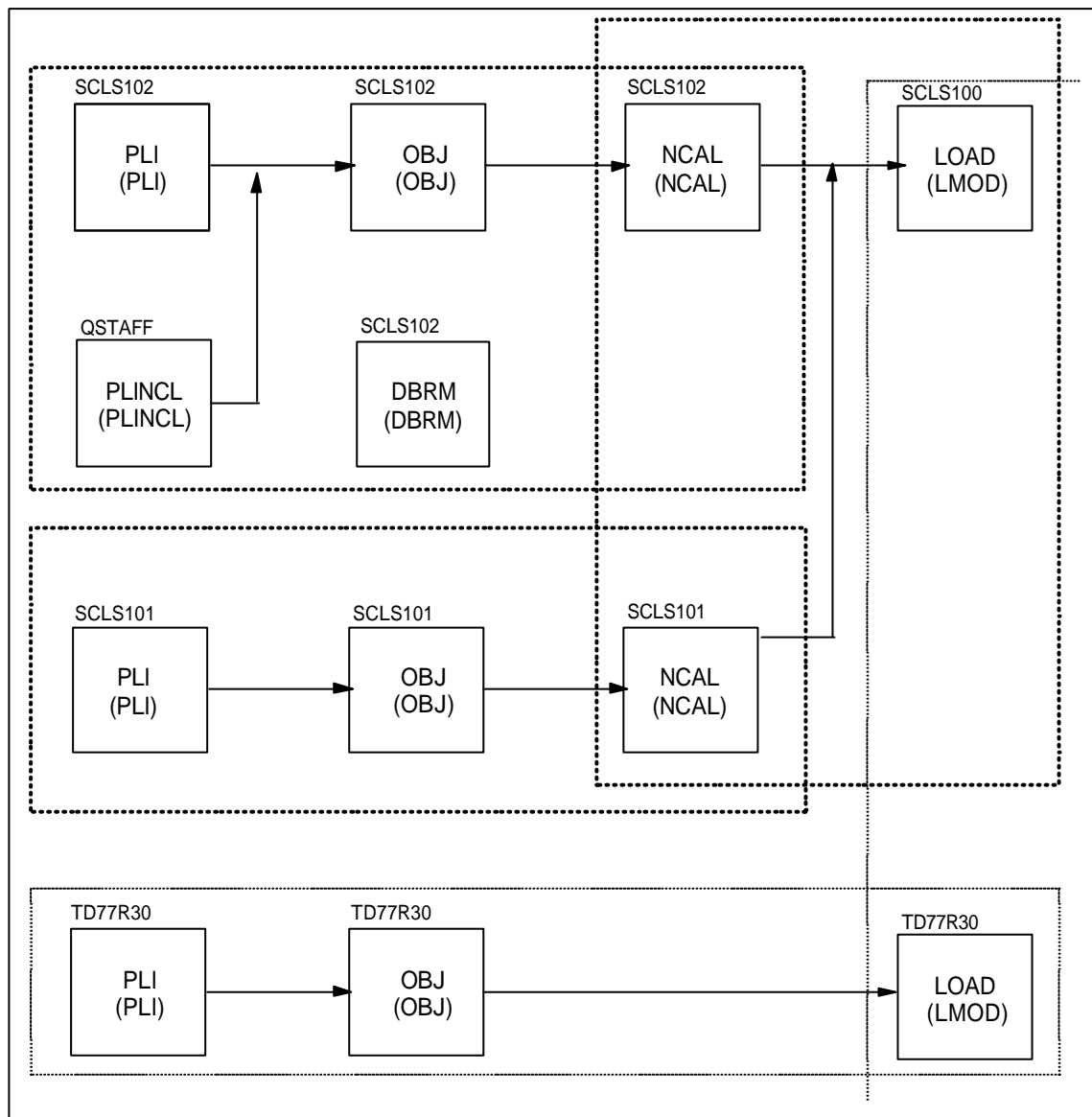
1つのアーキテクチャー・メンバーの中で、他のアーキテクチャー・メンバーや原始ソース・メンバーをまとめて指定することができる。

このアーキテクチャー・メンバーは、上位（HL）アーキテクチャー・メンバーといい、ARCHDEFライブラリーに作成する。

次ページの例では、アーキテクチャー・メンバーTD77LMODIは他の2つのアーキテクチャー・メンバーを指定している。

TD77LMODIを使用することにより、2つのロードモジュールを、一回の操作で「組立て」たり、「プロモート」することができる。

4.1 概要



ARCHDEF (TD77LMOD)

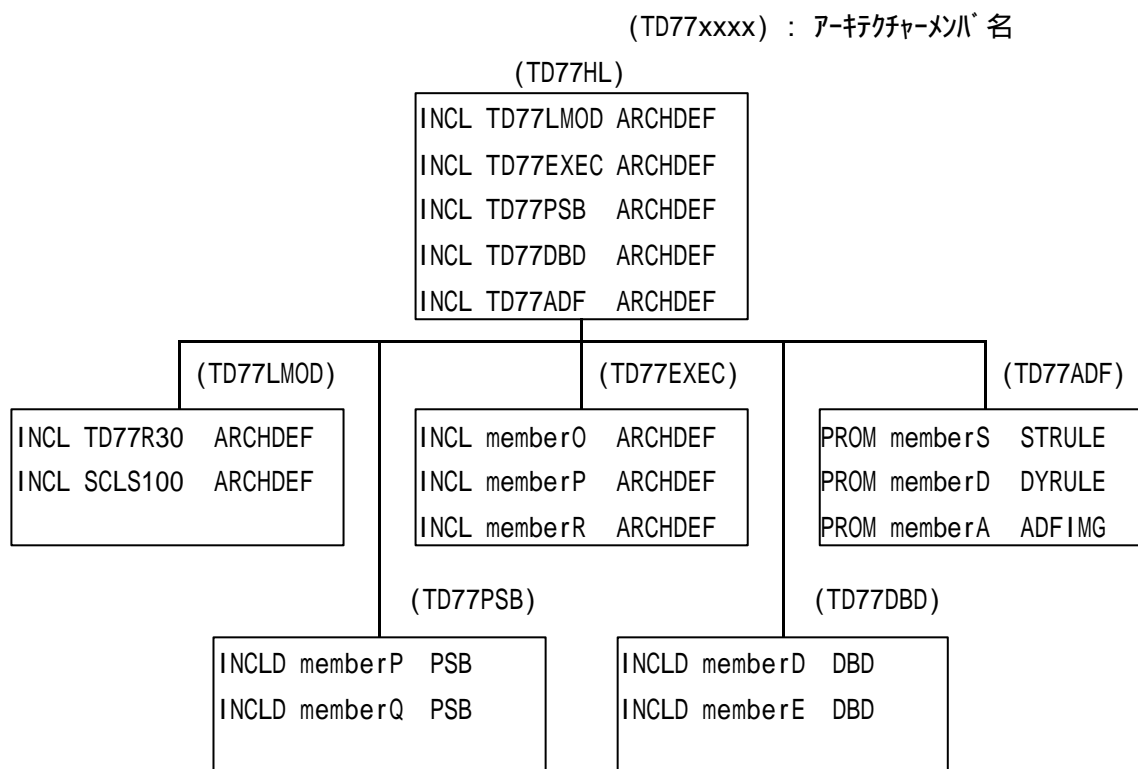
```

* ----- *
* MEMBER NEME      : TD77LMOD
* DEFINITION TYPE  : HL ARCHITECTURE DEFINITION
* DESCRIPTION      : DEFINE ALL BATCH EXECTABLE LOAD MODULE
* ----- *
INCL      TE77R30      ARCHDEF      * LOAD MODULE 1
INCL      SCLS100      ARCHDEF      * LOAD MODULE 2

```

4.1 概要

< アプリケーション TD77の全体像 >



アプリケーションを1つにまとめたアーキテクチャー・メンバー（ppppHL）を持つことが望ましいが、移管のタイミングを考慮すると、それは無理。

同じライブラリー・タイプ毎にまとめたアーキテクチャー・メンバーを作成する。



このメンバーを使用して、移管/メンテナンスも行う

4.2 ロードモジュール

1つメンバーの「組立て」のために、アーキテクチャー定義を作成しなければならないライブラリー・タイプ（LMOD，EXECなど）を、上位アーキテクチャー定義でまとめるときは、INCLキーワードで、アーキテクチャー・メンバーを指定する。

< サンプル >

ARCHDEF（TD77LMOD）

```
* ----- *
* MEMBER NEME      : ppppLMOD
* DEFINITION TYPE   : HL ARCHITECTURE DEFINITION
* DESCRIPTION       : DEFINE ALL BATCH EXECTABLE LOAD MODULE
* ----- *
INCL      memberA      ARCHDEF      * LOAD MODULE 1
INCL      memberB      ARCHDEF      * LOAD MODULE 2
```

ARCHDEFのアーキテクチャー定義を指定する。

またLMODまでは「組立て」ず、NCALまで「組立て」るメンバーを指定する時は、ARCHNCLのアーキテクチャー定義を指定する。

```
INCL      memberC      ARCHDEF      * NCAL MODULE
```

ARCHDEFのアーキテクチャー定義を指定する。

4.3 PSB/DBD

1つのメンバーの「組立て」のために、アーキテクチャー定義は作成しないが、原始ソース・メンバーから「組立て」によって出力されるライブラリー・タイプ（PSBLIB、DBDLIBなど）を、上位アーキテクチャー定義でまとめる時は、

INCLDキーワードで、原始ソース・メンバーを指定する

< サンプル >

ARCHDEF (TD77PSB)

```
* -----*
* MEMBER NEME      : ppppPSB
* DEFINITION TYPE   : HL ARCHITECTURE DEFINITION
* DESCRIPTION       : DEFINE ALL PSB
* -----*
INCLD      memberP      PSB      * PSB 1
INCLD      memberQ      PSB      * PSB 2
```



原始ソース・メンバーを指定する。

* APPENDIX *

- 付1 アーキテクチャ定義作成ツール
- 付2 AMSサービスで利用されるSCLMライブラリーのタイプ
- 付3 AMSサービスで利用される言語定義
- 付4 SCLM環境作成ツール一覧

付1. アーキテクチャー定義作成ツール

1. NCAL用 ARCHNCL 作成JCL

< 機能 >

“SOURCE”DDカードで指定したライブラリーの各メンバーに対し、NCAL用のアーキテクチャー定義を作成する。

ARCHNCL (memberA)

LKED	LINKN	
LOAD	memberA	NCAL
INCLD	memberA	RULES

<-パラメーターで指定されたタイプが入る>

< サンプル >

```
編集---TD77I.MODEL.JCL(GENNCAL)---01.01-----桁 001 072
コマンド==>                               画面移動==>CSR

.
.

// PEND
//*****
// *SOURCEからARCHNCL作成
//*****
//GENNCAL EXEC IKJEFT01
//SOURCE DD DSN=pppp.xxxLIBn.????,DISP=SHR      <=CHG YOUR SOURCE
//ARCHNCL DD DSN=pppp.xxxLIBn.ARCHNCL,DISP=OLD   <=CHG YOUR ARCHNCL
//SYSTSIN DD *
ISPATART PGM(GENNCAL) *
PARM(/????)
/*
```

- ・ TD77I.MODEL.JCL(GENNCAL) をコピーして、“SOURCE”と“ARCHNCL”のDDカードと“SYSTSIN”のPARMを変更する。
- ・ “SOURCE”のDDカードには、ARCHNCLのINCLDライブラリーを指定する。
- ・ ???には、RULES,PLI,ASMなどを指定する。

付1. アーキテクチャー定義作成ツール

2. SDTM・O/P-LOAD ARCHDEF作成JCL

< 機能 >

XREFファイルの情報をもとに、O-LOAD用/P-LOAD用のアーキテクチャー定義を作成する。

O-LOAD

ARCHDEF (ロード モジュール名)

ロード モジュール名=screenID + 0

LKED	SDTMLNK	
LOAD	ロード モジュール名	EXEC
INCL	O-PGM名	ARCHNCL
INCL	O-RULE 名	ARCHNCL
CMD	INCLUDE	SYSLIB (RWBRVUM)
CMD	ENTRY	O-PGM名
CMD	NAME	ロード モジュール名 (R)

P-LOAD

ARCHDEF (ロード モジュール名)

ロード モジュール名=screenID + P

LKED	SDTMLNK	
LOAD	ロード モジュール名	EXEC
INCL	I-PGM名	ARCHNCL
INCL	O-RULE 名	ARCHNCL
INCL	I -RULE 名	ARCHNCL
INCL	O-RULE 名	ARCHNCL
INCL	B-RULE 名	ARCHNCL
INCL	E-RULE 名	ARCHNCL
INCL	飛び先画面のO-PGM名	ARCHNCL
INCL	飛び先画面のO RULE 名	ARCHNCL
CMD	INCLUDE	SYSLIB (RWBRVUM)
CMD	ENTRY	O-PGM名
CMD	NAME	ロード モジュール名 (R)

付1. アーキテクチャー定義作成ツール

< サンプル >

```

編集---TD77I.MODEL.JCL(GENSDTM)---01.01-----桁 001 072
コマンド==>                                     画面移動==>CSR

.
.

// PEND
//*****
//* XREF から ARCHNCL作成
//*****
//STEP      EXEC IKJEFT01
//SYSUT1 DD  DSN=TD67I.TEST.XREF,DISP=SHR          <=CHG YOUR XREF
//SYSUT2 DD  DSN=&&XREF,DISP=(,PASS),
//           SPACE=(TRK,(10,20),RLSE)
//           UNIT=3380,
//           DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//SYSTSIN DD  *
SELECT MEMBER=(@@@@@@@@)
/ *
//GENSDTM DD  DSN=&&XREF,DISP=(OLD,DELETE)
//XREF      DD  DSN=pppp.xxxLIBn.ARCHDEF,DISP=OLD   <=CHG YOUR ARCHDEF
//SYSTSIN DD*
        ISPSTART PGM(GENSDTM)
/*

```

- ・ TD77I.MODEL.JCL(GENSDTM) をコピーして、“SYSUT1”と“ARCHNCL”のDDカードと“SYSDIN”の MEMBERを変更する。
- ・ “SYSUT1”のDDカードには、SDTMのXREFライブラリーを指定する。
- ・ “SYSDIN”のMEMBERの@@@@@@@@には、XREFのメンバー名を入れる。
- ・ XREFがシーケンシャルの場合には、POTOPSのステップ名を削除して、“XREF”のDDカードを一時データセットからXREFライブラリーに変更する。
- ・ 作成されるアーキテクチャー定義は、XREFファイルの情報をもとに作成される。
PLI/ASMのサブルーチンをリンクする場合やDB2を使用する場合には、O/P-LOADともアーキテクチャー定義に以下を追加する必要がある。

PLIサブルーチンをリンクするとき

```
INCL    PLIサブルーチン名    ARCHNCL
```

ASMサブルーチンをリンクするとき

```
INCL    PLIサブルーチン名    ARCHNCL
```

DB2を使用するとき

```
INCL    PLIサブルーチン名    ARCHNCL
```

付2. AMSサービスで使用するSCLMライブラリーのタイプ

	ライブラリー・タイプ (3rd Qualifier)	内容	選択基準	
ソース	PLI	PLIソース	M	必須〔PLIの場合〕
	PLINCL	PLIマクロ(%INC)	0	PLIマクロ(%INC)がある場合
	ASM	ASMソース	0	ASMソースがある場合
	MAC	ASMマクロ(%INC)	0	ASMマクロ(%INC)がある場合
	C	C370ソース	0	C370ソースがある場合
	CINCL	C370ヘッダーファイル	0	C370ヘッダーファイルがある場合
	COBOL	COBOLソース	0	COBOLソースがある場合
	COBINCL	COBOLマクロファイル	0	COBOLマクロがある場合
	RULES	SDTMルールプログラム	0	SDTMルールがある場合
	PSB	PSBソース	0	PSBソースがある場合
	DBD	DBDソース	0	DBDソースがある場合
	STRULE	ADF静的規則	0	ADFソースがある場合
	DYRULE	ADF動的規則	0	ADFソースがある場合
	ADFIMG	ADF画面イメージレイアウト	0	ADFソースがある場合
中間生成物	OBJ	オブジェクト・モジュール	M	必須
	PLKOBJ	PRELINKオブジェクト・モジュール	0	C370ソースがある場合
	PSBOBJ	PSBオブジェクト・モジュール	0	PSBをALIAS使用する場合
	NCAL	NACLモジュール	0	NCALがある場合
	DBRM	DB2 DBRM	0	DB2プログラムがある場合
ロード	LMOD	バッチ実行ロードモジュール	0	バッチがある場合
	LMODx	バッチ実行ロードモジュール	0	注)
	EXEC	オンライン実行ロードモジュール	0	IMSオンラインがある場合
	EXECx	オンライン実行ロードモジュール	0	注)
	ISPLLIB	TSOオンライン実行ロードモジュール	0	TSOオンラインがある場合
	ISPLLIBx	TSOオンライン実行ロードモジュール	0	注)
	PSBLIB	PSBロードモジュール	0	PSBソースがある場合
	DBDLIB	DBDロードモジュール	0	DBDソースがある場合
定義	ARCHNCL	アーキテクチャ定義(NCAL用)	0	NCALがある場合
	ARCHDEF	アーキテクチャ定義	M	必須(PLIの場合)
	ARCHPLK	アーキテクチャ定義(C370用)	0	C370ソースがある場合

M: 必須 / 0: オプション

注) LMODxなどはロードモジュールライブラリーを分けて持つ必要がある場合に使用する。

付3. AMSサービスで使用する言語定義

1. 各言語定義のプロトタイプは、「TD771.MODEL.SOURCE」にあるので、それらを参照する事。
(メンバー名：@xxxxx)
2. 言語名はソースまたはアーキテクチャー定義に関しては編集時に指定(プロンプトOR SPROFコマンド)し、ロードに関してはアーキテクチャー定義内でLKEDステートメントで指定する。
3. 異なるバージョンのコンパイラを使用する必要がある場合は、AMS技術にお問い合わせ下さい。各プロジェクトで独自に変更を行うと、ソフトウェアのリリース・アップ時に問題となる場合があります。

ライブラリタイプ	言語定義名	SCLM管理外のシステムライブラリ	ステップ	使用オプション
ソース	PLI2 (NON-DB2 V2)	SYS1.CRD/SYS1.TESTCRD SYS1.URD1, SYS1.DCL SYS1.CTRL, SYS1.CTRL2 SYS1.UCTRL	CRD/EXPAND	P
			COMPILE	AG,A(S),F(1),OPT(2),NEST,M,S,OP,CMPAT(V2)
	PLIM(LE)	SYS1.CRD/SYS1.TESTCRD SYS1.URD1,SA00.DS.DCL SYS1.CTRL2,SYS1.DCL SYS1.CTRL,SYS1.UCRL	CRD/EXPAND	P
			COMPILE	NIS,A,X,AG,M,MAR(2,72,1) NIS,A,X,AG,M,MAR(2,72,1),GOSTMT AG,A(S),F(1),OPT(2),NEST,M,S,OP,CMPAT(V2)
	PLIC (CICS V4)	SYS1.CRD/SYS1.TESTCRD SYS1.URD1, SYS1.DCL SYS1.CTRL, SYS1.CTRL2 SYS1.UCTRL CICS.V4R1MO.SDFHPLI CICS.V4R1MO.SDFHMAC	CRD/EXPAND	P
			TRANSLATE	CISC,EDF,DLI
			COMPILE	AG,A(S),F(1),OPT(2),NEST,M,S,OP
	PLI2DB2 (DB2)	SYS1.CRD/SYS1.TESTCRD SYS1.URD1, SYS1.DCL SYS1.CTRL, SYS1.CTRL2 SYS1.UCTRL	CRD/EXPAND	P
			PRECOMPILE	HOST(PLI),SOURCE
			COMPILE	AG,A(S),F(1),OPT(2),NEST,M,S,OP

付3. AMSサービスで使用する言語定義

ライブラリタイプ	言語定義名	SCLM管理外のシステムライブラリ	ステップ	使用オプション
ソース	PLI (続き)	PLIMDB2(LE) SYS1.CRD, SYS1.TESTCRD SYS1.URD1 SA00.DS.DCL, SYS1.CTR2, SYS1.DCL, SYS1.CTRL, SYS1.UCTRL, SYS1.SCEESAMP	CRD/EXPAND	P
			PLI MACRO EXPANSION	MACRO, NOSYNTAX, MDECK, NOIN SOURCE, NOSOURCE
			PRECOMPILE	HOST(PLI), SOURCE
			COMPILE	AG, A(S), F(1), OPT(2), NEST S, OP
	PLICDB2 (CICS-DB2 V4)	SYS1.CRD, SYS1.TESTCRD SYS1.URD1, SYS1.DCL SYS1.CTRL, SYS1.CTRL2 SYS1.UCTRL CICS.V4R1M0.SDFHPL1 CICS.V4R1M0.SDFHMAC	CRD/EXPAND	P
			PRECOMPILE	HOST(PLI), SOURCE
			TRANSLATE	CICS, EDF, DLI
			COMPILE	AG, A(S), F(1), OPT(2), NEST S, OP
	PLIMCDB(LE)	SYS1.CRD, SYS1.TESTCRD SYS1.URD1, SA00.DS.DCL SYS1.CTRL2, SYS1.DCL SYS1.CTRL, SYS1.UCTRL CICS.V4R1M0.SDFHPL1 CICS.V4R1M0.SDFHMAC SYS1.SCEESAMP	CRD/EXPAND	P
			PLI MACRO EXPANSION	MACRO, NOSYNTAX, MDECK, NOIN SOURCE, NOSOURCE
			PRECOMPILE	HOST(PLI), SOURCE
			TRANSLATE	CICS, EDF, SP
			COMPILE	AG, A(S), F(1), OPT(2), NEST S, OP
	RULES (NON-DB2 V2)	SYS1.CRD, SYS1.TESTCRD SYS1.URD1, SYS1.DCL SYS1.CTRL, SYS1.CTRL2 SYS1.UCTRL ECS.COMM.ECSINCL ECS.COMM.PL I ECS.COMM.PL INCL	CRD/EXPAND	P
			COMPILE	AG, A(S), F(1), OPT(2), NEST M, S, OP
	RULESM(LE)	SYS1.CRD, SYS1.TESTCRD SYS1.URD1 SYS1.CTRL2, SYS1.DCL SYS1.CTRL, SYS1.UCTRL ECS.COMM.ECSINCL.R10 ECS.COMM.PL I.R10 ECS.COMM.PL INCL.R10	CRD/EXPAND	P
			COMPILE	AG, A(s), F(1), OPT (2), NEST, M, S, OP
	RULEDB2 (DB2)	SYS1.CRD, SYS1.TESTCRD SYS1.URD1, SYS1.DCL SYS1.CTRL, SYS1.CTRL2 SYS1.UCTRL ECS.COMM.ECSINCL ECS.COMM.PL I ECS.COMM.PL INCL	CRD/EXPAND	P
			PRECOMPILE	HOST(PLI), SOURCE
			COMPILE	AG, A(S), F(1), OPT(2), NEST S, OP, CMPAT(V2)

付3. AMSサービスで使用する言語定義

ライブラリタイプ	言語定義名	SCLM管理外のシステムライブラリ	ステップ	使用オプション
PLI (続き)	RULESDBM (LE)	SYS1.CRD, SYS1.TESTCRD SYS1.URD1 SYS1.CTRL2, SYS1.DCL SYS1.CTRL, SYS1.UCTRL ECS.COMM.ECSINCL.R10 ECS.COMM.PLI.R10 ECS.COMM.PLINCL.R10	CRD/EXPAND	P
			PRECOMPILE	HOST(PLI), SOURCE
			COMPILE	AG, A(S), F(I), OPT(2), NEST, SOP, CMPAT(V2)
ソース	C370 (NON-DB2)	EDC.V2R1M0.SEDCHDRS	COMPILE	XREF, LANG(SAAL2), SO, OPT, TEST (ALL), MAR(1,72), NOTERM, FL(I), SHOW
			TRANSLATE	CICS, EDF
			COMPLIE	RENT, SOURCE
			PREPROCESS	HOST®, SOURE
	C370DB2 (DB2)	EDC.V2R1M0.SEDCHDRS SYS1.DCL	COMPILE	XREF, LANG(SAAL2), SO, OPT, TEST (ALL), MAR(1,72), NOTERM, FL(I), SHOW
			PRECOMPILE	HOST®, SOURCE
			TRANSLATE	CICS, EDF
			COMPILE	RENT, SOURCE
	C370DB2 (CICS-DB2)	EDC.V2R1M0.SEDCHDRS CICS.V4R1M0.SDFHC370 CICS.V4R1M0.SDFHMAC SYS1.DCL	PRECOMPILE	HOST®, SOURCE
			TRANSLATE	CICS, EDF
			COMPILE	RENT, SOURCE
			COMPILE	RENT, SOURCE
I	ASM	ASM	ASSEMBLE	XREF(SHORT), LINECOUNT(75), OBJECT
	COBOL	COBMVS	COMPILE	OBJECT XREF, LIB, APOST, NODYNAM, LIST, NONUMBER, NOSEQ
			COMPILE	OBJECT XREF, LIB, APOST, NODYNAM, LIST, NONUMBER, NOSEQ
	PSB	PSB	PSBGEN	XREF, LIST, DCBS
	DBD	DBD	DBDGEN	XREF, LIST
	STRULE	STRULE	カスタイズ 不要 (ADFの管理にのみ使用)	
	DYRULE	DYRULE		
	ADFIMG	ADFIMG		
	全てのソース・ タイプ	TEXT	カスタイズ 不要 (ソース管理にのみ使用)	

付3. AMSサービスで使用する言語定義

	ライブラリタイプ	言語定義名	SCLM管理外のシステムライブラリ	ステップ	使用オプション
オブジェクト	PLKOBJ	PLINK	SYS1.SIBMLINK EDC.V2R1M0.SEDMSGSE(EDCMSGRE)	PRELINK	NO-OPTION
ロード	NCAL	LINKN	SYS1.PLIBASE, SYS1.SIBMBASE SYS1.PLITASK, SYS1.COB2LIB ISPF.SISPLOAD IMSVS.RESLIB DSN.DSNLOAD CICS.V4R1M0.SDFHLOAD DMS001A.LMOD2 DMS001A.LMOD	LKED/370	DCBS, MAP, LET, LIST, XREF, NCAL, REUS
		LINKM (LE)	SA00.DS.NCAL.LE SYS1.SCEELKED, SYS1.SCEERUN DCP.DCPLIB, ISPF.SISPLOAD IMSVS.RESLIB, DSN.DSNLOAD CICS.V4R1M0.SDFHLOAD DMS001A.LMOD2, DMS001A.LMOD	LKED/370	LET, LIST
	PSBLIB	PSBLNK (ALIAS使用)		PSB LINK	XREF, LIST
	LMOD or EXEC	LINK2 (V2)	SYS1.PLIBASE, SYS1.SIBMBASE SYS1.PLITASK, SYS1.COB2LIB ISPF.SISPLOAD IMSVS.RESLIB DSN.DSNLOAD CICS.V4R1M0.SDFHLOAD DMS001A.LMOD2, DMS001A.LMOD SYS1.SCEELKED	LKED/370	DCBS, MAP, LET, LIST, XREF
		LINKM (LE)	SA00.DS.NCAL.LE SYS1.SCEELKED SYS1.SCEERUN, DCP.DCPLIB ISPF.SISPLOAD IMSVS.RESLIB, DSN.DSNLOAD CICS.V4R1M0.SDFHLOAD DMS001A.LMOD2, DMS001A.LMOD	LKED/370	LET, LIST

付3. AMSサービスで使用する言語定義

	ライブラリ タイプ	言語定義名	SCLM管理外のシステムライブラリ	ステップ	使用オプション
デ ー タ	LMOD or EXEC	DB2LNK2 (NON-IMS DB2 V2)	SYS1.PLIBASE, SYS1.SIBMBASE SYS1.PLITASK SYS1.COB2LIB ISPF.SISPLoad DSN.DSNLOAD, IMSVS.RESLIB CICS.V4R1M0.SDFHLOAD DMS001A.LMOD2, DMS001A.LMOD	LKED/370	DCBS, MAP, LET, LIST, XREF
		DB2LINKM (LE)	SA00.DS.NCAL.LE SYS1.SCEELKED SYS1.SCEERUN, DCP.DCPLIB ISPF.SISPLoad, DSN.DSNLOAD MSVS.RESLIB, DSN.DSNLOAD CICS.V4R1M0.SDFHLOAD DMS001A.LMOD2, DMS001A.LMOD	LKED/370	DCBS, MAP, LET, LIST, XREF
		SDTMLNK(V2)	SYS1.PLIBASE, SYS1.SIBMBASE SYS1.PLITASK, SYS1.COB2LIB ISPF.SISPLoad, IMSVS.RESLIB DSN.DSNLOAD CICS.REL21.LOADLIB DMS001A.LMOD2 DMS001A.LMOD ECS.COMM.AUTOLOAD ECS.COMM.LOAD	LKED/370	DCBS, MAP, LET, LIST, XREF, REUS, AMODE=31, RMODE=24
		SDTMLNM (LE)	SYS1.SCEELKED, SYS1.SCEERUN ISPF.SISPLoad, IMSVS.SDFSRESL DSN, DSNLOAD ECS.COMM.AUTOLOAD.R10 ECS.COMM.LOAD.R10	LKED/370	DCBS, MAP, LIST, XREF, REUS, AMODE=31, RMODE=24
	ARCHNCL	ARCHDEF			
	ARCHDEF				
	ARCHPLK				

→ アーキテクチャ定義

付4．SCLM環境作成ツール一覧

各D/Sのアロケーション又は、プロジェクト定義(ロード)作成用のアセンブル・リンクのプロトタイプJCLは、「TD771.MODEL.JCL」にあるので、それらを参照する事。

また、全てSMSデータセットのため、VOLUMEを指定しなくてよい。

1．テスト用ライブラリー (RELLIB以外)のアロケーション (RELLIBはサービス管理に依頼)

各ライブラリーの属性は下図の通り。プロトタイプJCL (LIBRARY) を参考にして作成する。

タイプ	RECFM	LRECL	BLKSZE
ソース	FB	80	3120
目的モジュール	FB	80	3120
ロードモジュール	U	0	13030
アーキテクチャー定義	FB	80	3120
DBRM	FB	80	3120

なお、必要となるスペース量は、次の公式を目安として計算する。とのライブラリーも、容量の60%程度が満たされるように想定する。

M = メンバー数
 T = 必要となるトラック数 (3390)
 B = 必要となる登録簿ロック数



T = $M \times 1.2 / 0.6$ <プログラムソース、目的、ロード、DBRM>
 T = $M \times 1.2 / 0.6$ <ADFソース、PSB/DBソース・ロード>
 T = $M \times 1.2 / 0.6$ <アーキテクチャー定義>

B = $M / 5 / 0.6$

付4．SCLM環境作成ツール一覧

2. プロジェクト定義データセット (PDS)

各データセットの属性は下図の通り。プロトタイプJCL (PROJDEFS) を参考にして作成する。

データセット名	RECFM	LRECL	BLKSZE
プロジェクト定義D/S (ロード) pppp . PROJDEFS . LOAD	U	0	6144
プロジェクト定義D/S (ソース) pppp I . PROJDEFS . SOURCE	FB	80	3120

なお、必要となるスペース量は、次の公式を目安として計算する。

M = メンバー数
T = 必要となるトラック数 (3380)
B = 必要となる登録簿簿ロック数

↓

T = M+2	1メンバー当たり1トラック
T = (M+2) / 5	1メンバー当たり0.2ブロック

3. 会計データセットのアロケーション (VSAM)

各データセットの属性は下図の通り。プロトタイプJCL (ACCOUNT) を参考にして作成する。

データセット名	LRECL	BLKSZE
会計データセット(本番用) ppppIV . ACCOUNT . FILE	264	8K
会計データセット(テスト用) ppppIVT . ACCOUNT . FILE	264	8K

なお、必要となるスペース量は、次の表を目安として計算する。

制御下にあるメンバーの数	シリンダー数 (3390)
1,000	2.5
10,000	5.0

付4．SCLM環境作成ツール一覧

4．監査制御D/S，バージョン区分D/Sのアロケーション

各データセットの属性は下図の通り。プロトタイプJCL（AUDIT/VERSION）を参考にして作成する。

データセット名	ORG	RECFM	LRECL	BLKSZE
監査制御データセット ppppIVT . AUDIT . FILE	VSAM		264	8K
バージョン区分データセット ppppI . XxxLIB . SOURCE . VERSION	PDS	VB	259	9069

なお、バージョン区分データセット作成に必要なスペース量は、少なくともSCLM管理下にあるソース・ライブラリー(PLI，RULES，etc...)を合計したものが必要となる。

付4．SCLM環境作成ツール一覧

< プロトタイプ JCL 一覧 >

使 用 目 的	JCL メンバー名
SCLMテスト用ライブラリーのアロケーション (DEVLIB1,DEVLIB2,TSTLIB,FXLIB,UTLLIB)	TD77I,MODEL.JCL (LIBRARY)
プロジェクト定義データセットの作成「PDS」 (ロード or ソース)	TD77I,MODEL.JCL (PROJDEFS)
会計情報データセットの作成 「VSAM」 (本番orテストライブラリー用)	TD77I,MODEL.JCL (ACCOUNT)
会計制御データセットの作成 「VSAM」	TD77I,MODEL.JCL (AUDIT)
バージョン区分データセットの作成 「PDS」	TD77I,MODEL.JCL (VERSION)
プロジェクト定義のアセンブル/リンク 「プロジェクト定義ロード作成」	TD77I,MODEL.JCL (ASMLKJCL)

End of Document

Last Updated:06/30/2000