

アプリケーション・ライフサイクル管理

ASG-Life Cycle Managerご紹介

LCMの主な特徴:

コンポーネント管理

COBOL、アセンブラー、JCL、COPYBOOK、実行ファイルなどの様々なタイプのオブジェクトをLCMに取り込み保持、管理

バージョン管理/変更履歴/アーカイブ

LCM管理下の全オブジェクト変更の証跡記録(ユーザー、変更日など)を管理変更したオブジェクトはデルタ(差分)技術を使用してバージョン管理

ライフサイクル機能

開発プロセス、オブジェクト変更を制御して、認証機能で認可された変更を確実にリリース柔軟性を持ち、他製品のライフサイクルへも対応が可能

依存関係とビルドの管理

コンパイル時に補助的なオブジェクトがコンパイルされる様に、ビルドプロセスを管理

影響調査

オブジェクト変更後、他のオブジェクトに与える影響を調査

レポート機能

開発の為に必要な様々なレポート機能を提供

変更レポート、注釈テキスト、メンバータイプレポート、構成レポート、リスト差分レポート、関係性レポート

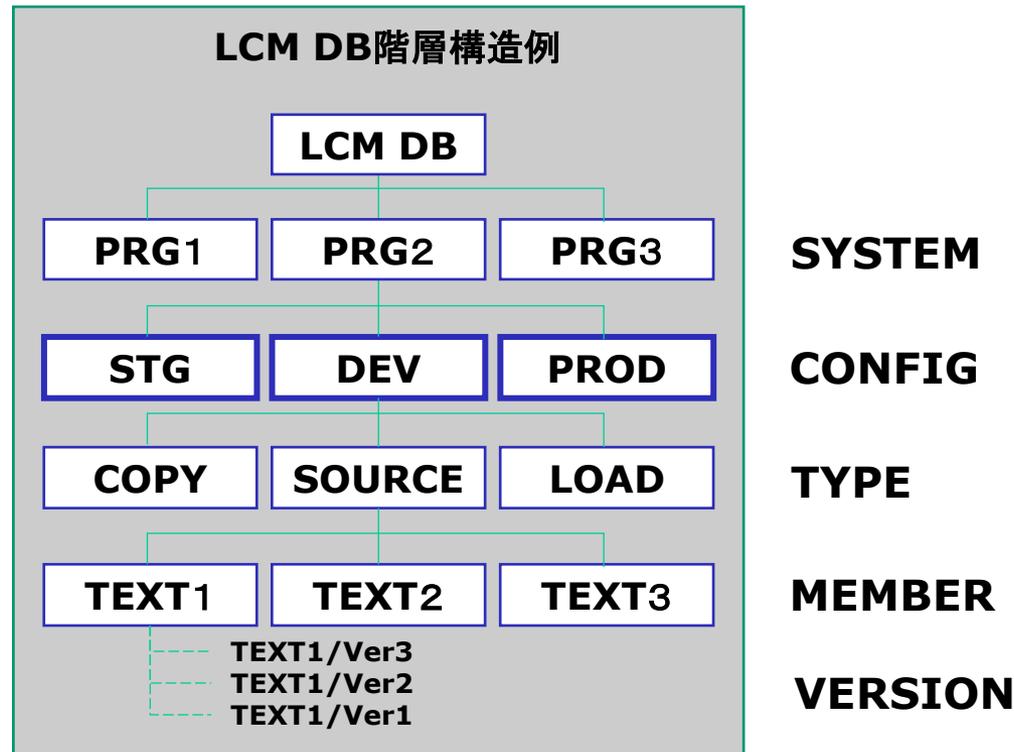
- **LCMデータベース**

データベースの構成について

- コンポーネント管理とバージョン管理
- パッケージ管理
- 分析/構築(ビルド)機能
- 必要に応じてカスタマイズ

LCMデータベース構造

LCM DB内は階層構造になっており、4つの要素(SYSTEM、CONFIG、TYPE、MEMBER)で構成されており、修正履歴をVERSIONとして管理します。
LCM DBには、LCM管理者が設定したユーザーやLCMコンポーネントのみがアクセスする事ができます。



ASG is the premier source for understanding, managing, and extending existing systems.

- LCM-DB
- コンポーネント管理とバージョン管理
- パッケージ管理
- 分析/構築(ビルド)機能
- 必要に応じてカスタマイズ

コンポーネント管理

- コンポーネントはタイプとメンバー名によって定義されます：
データベース/システム/コンフィグレーションの配下
- タイプはそれぞれの種類で指定できます：
SOURCE, COPY、LOADなどをタイプとして指定し実際のPDSファイルがひも付きます。
- メンバー名
各タイプには紐づいた**PDS**ファイルのメンバーが割り当てられます。

バージョン管理

- LCMはデルタ(差分)技術を使用して各バージョンを格納します。(基本+差分)
- バージョンは、インポート時に作成されデルタ(差分)が記録されますが、参照時には全体が表示されます。
- バージョンを特定する履歴を記録する:
連番、変更名称 (パッケージ管理に使用)、インポートしたユーザID
インポートした日付/時間

- **LCM-DB**

- コンポーネント管理とバージョン管理

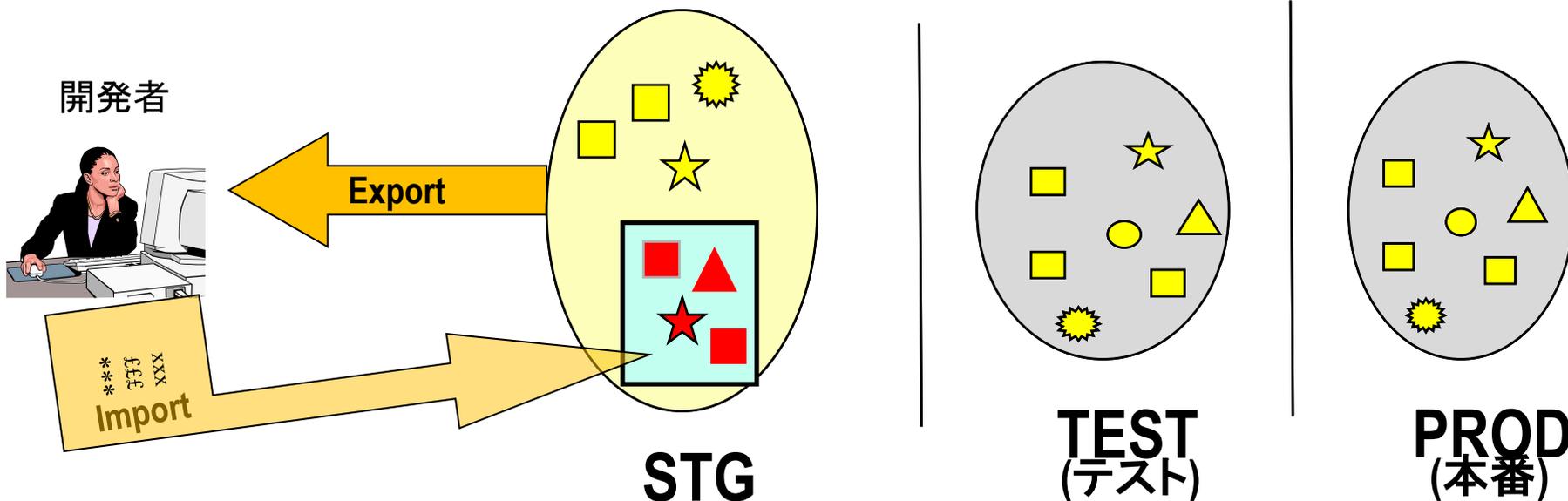
- **パッケージ管理**

パッケージはLCMがDEV、TEST、PRODのように構成を通して変更のグループを動かす方法です。

- 分析/構築(ビルド)機能

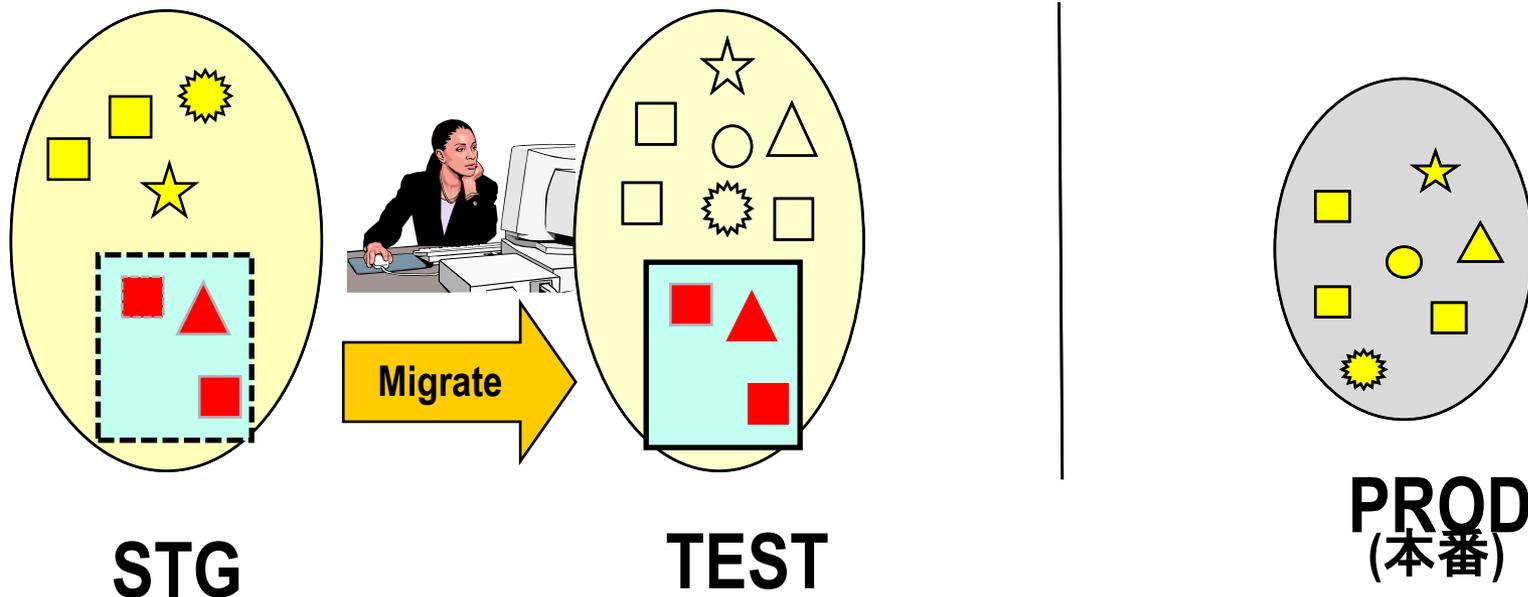
- 必要に応じてカスタマイズ

パッケージ - 作成



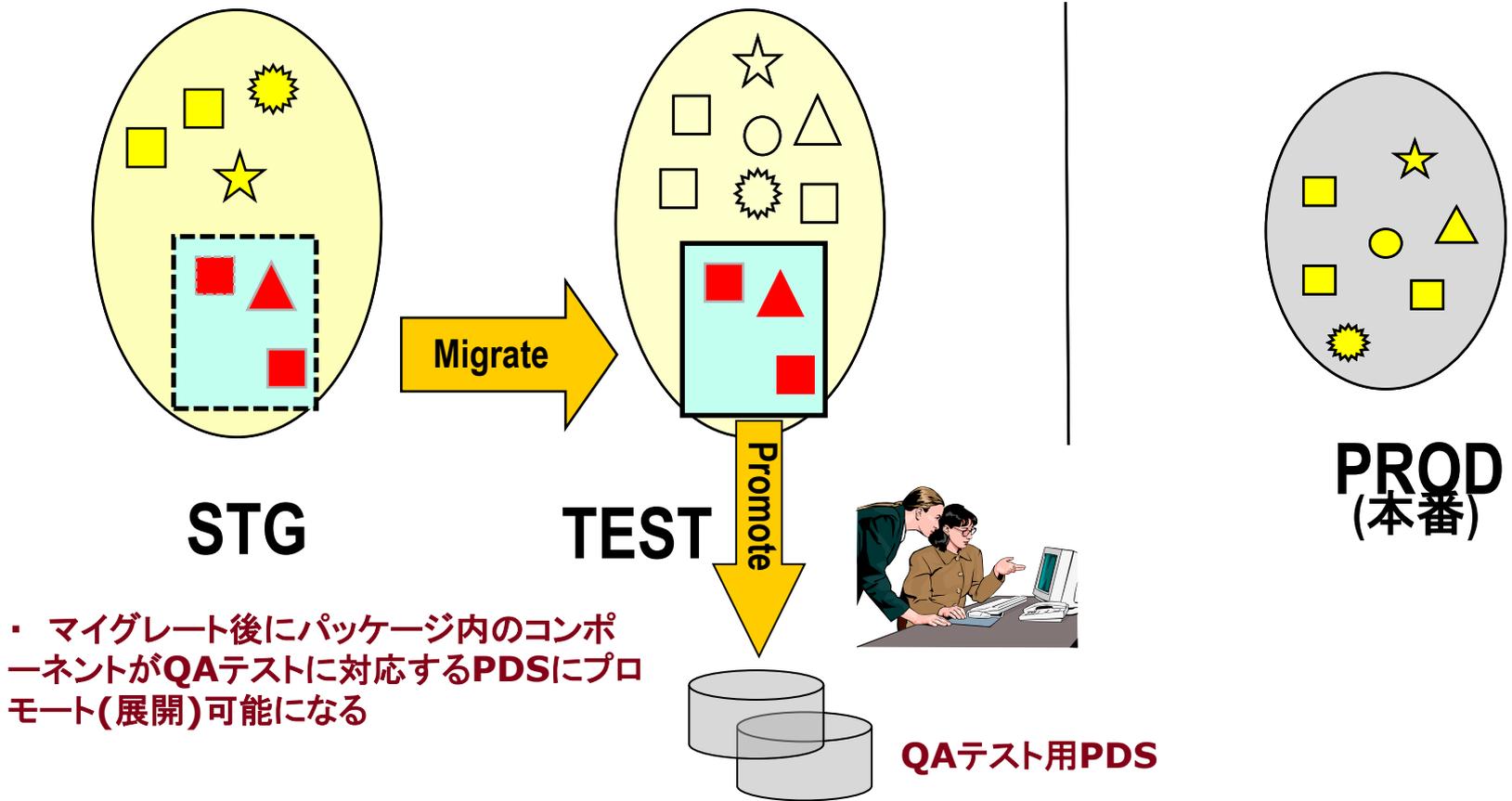
- ・ コンポーネントには、**LCM**から開発者によりエクスポートされ、変更され、インポートして戻されます
- ・ これらのコンポーネントはプログラムソース、**COPYBOOK**、**JCL**、**PROC**などです
- ・ 開発者が変更を行って、単体テストして、次にパッケージ変更名を使用して**LCM**へ変更をインポートします
- ・ エクスポートした人がそれらをインポートするまで、他の誰もこれらのコンポーネントをチェックアウトできません
(並行開発は可能です)

パッケージ – 移動(Migrate)



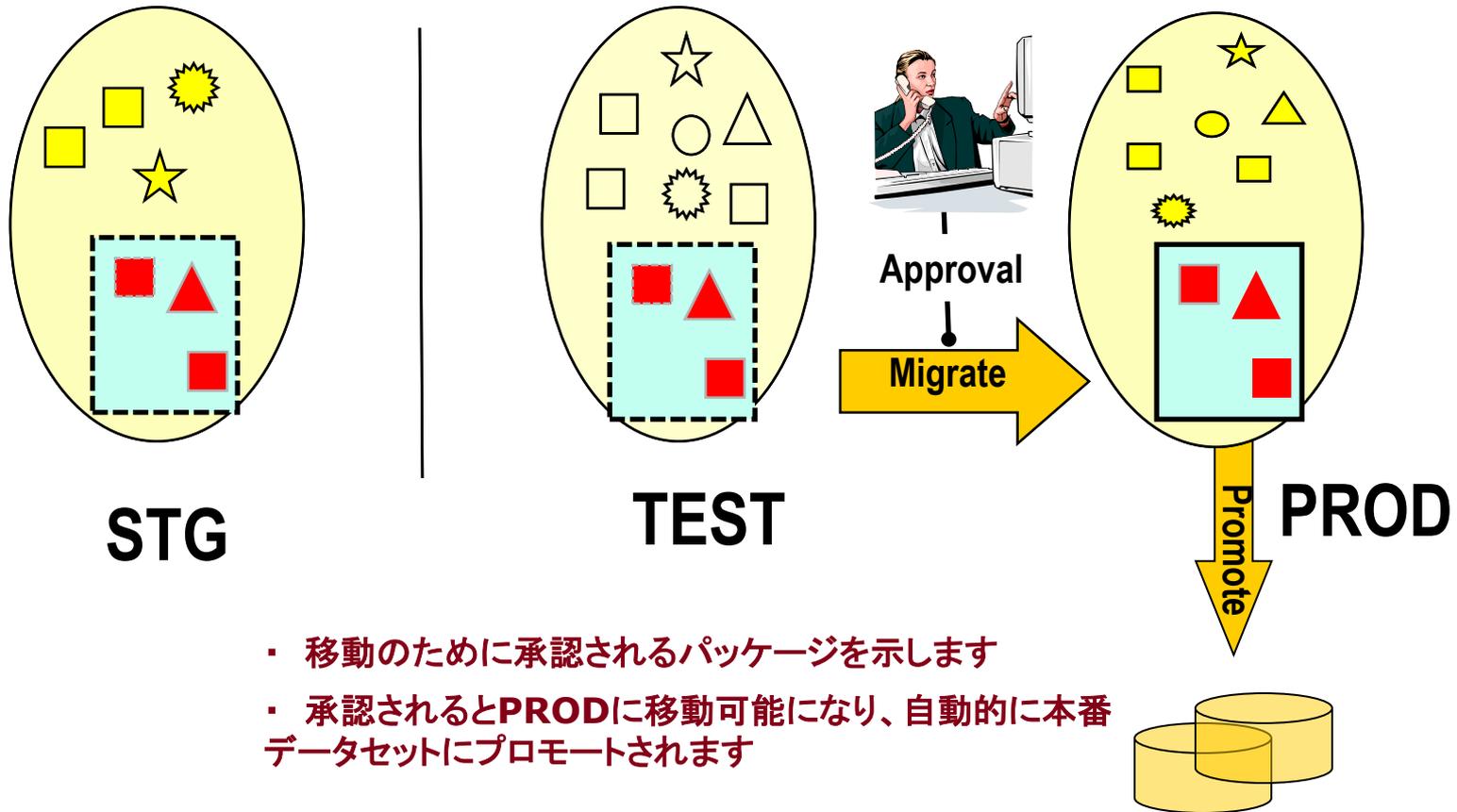
- ・ パッケージ内のコンポーネントが修正されて、STGからTESTに移動する

パッケージ – プロモート(展開)



- ・ マイグレート後にパッケージ内のコンポーネントがQAテストに対応するPDSにプロモート(展開)可能になる

パッケージ – 承認で移動



- ・ 移動のために承認されるパッケージを示します
- ・ 承認されると**PROD**に移動可能になり、自動的に本番データセットにプロモートされます

パッケージ - メニュー

```

----- ALLEN SYSTEM GROUP -----LCM ACTIVE ---
                PACKAGE FUNCTIONS                LCM.4
*****
COMMAND ==>

DATA BASE: DEMO
SYSTEM   : DEMO

      S   SET UP   - Set up for package functions
      1   ASSIGN  - Create package, define components, or edit notes
      2   MANAGE  - Create package, define life cycle, or edit notes
      3   DELETE  - Remove a package definition
      4   APPROVE - Approve or cancel approval for migration
      5   MIGRATE - Move package components along migration paths
      6   FALLBACK - Reject a previous migration
      7   PROMOTE - Export package components to host data sets
      8   REPORT  - List package information and user access

PACKAGE ID           ==> DVDEMO  (* or embedded * allowed with selection list)
PACKAGE SELECTION LIST (Y/N) ==> N

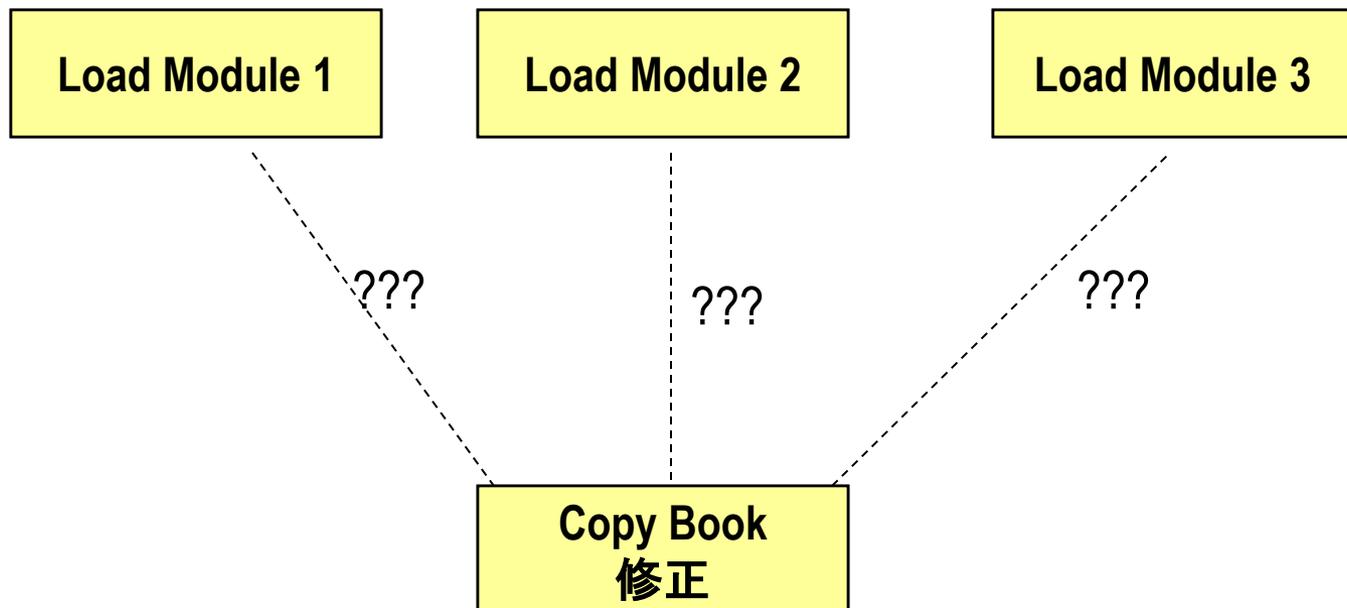
```

- LCM-DB
- コンポーネントとバージョン管理
- パッケージ管理
- **分析/構築(ビルド)機能**
依存関係を分析しビルドを正しく行う機能
- 必要に応じてカスタマイズ

依存管理

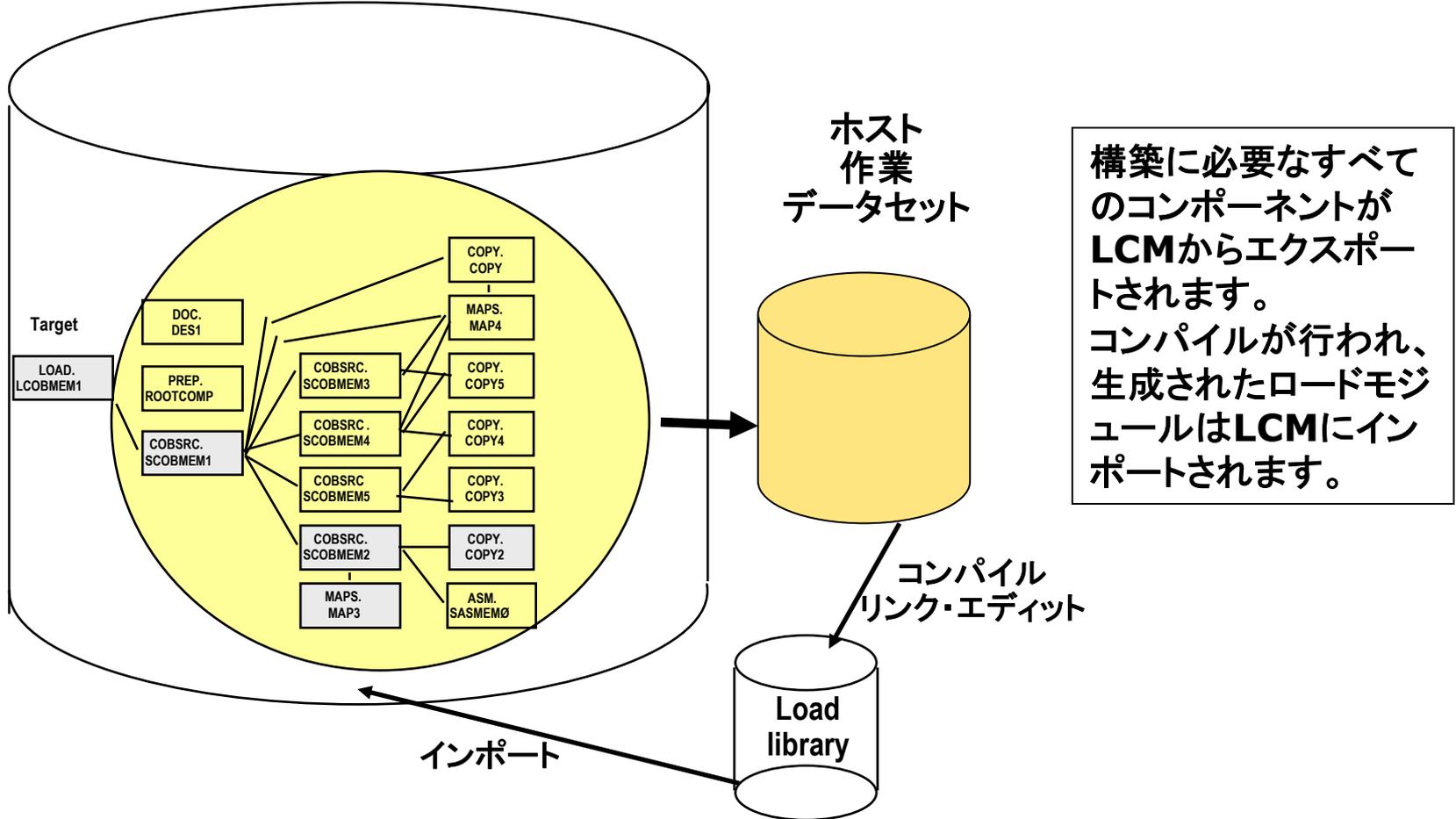
- 各コンポーネントをビルドするために、LCMは他のコンポーネントとのリンク情報(Copybook、Call、Linkなど)を確認します。
- ビルドされるコンポーネントがおよぼす他コンポーネントへの影響分析を行います。
- 分析結果は作業領域に格納されて、構築(ビルド)機能で使用されます。
- 修正を行う前に、その影響を知ることができます。
COPYBOOKかソース・プログラムの修正が、いずれかのロードモジュールに影響を与えるかユーザに伝えます。

依存関係 - 分析



どのコンポーネントが影響を受けるか? (構築必要か)
この修正によって? ...

依存 - 構築(ビルド)



構築に必要なすべてのコンポーネントが LCM からエクスポートされます。コンパイルが行われ、生成されたロードモジュールは LCM にインポートされます。

依存と構築(ビルド)メニュー

```
----- ALLEN SYSTEM GROUP -----LCM ACTIVE ---
                        DEPENDENCY AND BUILD FACILITY                LCM.3
*****
OPTION      ===>

SYSTEM NAME ===> DEMO
WORK AREA   ===> DEMO                (Blank or pattern for selection list)

1      SET UP          - Prepare for analysis and build
2      ANALYZE         - Determine dependency networks
3      BUILD           - Controlled build and target stamp utilities
4      DEPENDENCY REPORTS - Generate dependency reports
5      SCANNER UTILITIES - Delete scanner info or explicitly invoke scan
```

- コンポーネントとバージョン管理
- パッケージ管理
- 分析/構築(ビルド)機能
- 必要に応じてカスタマイズ

LCMインタフェース

- Naturalインタフェース
ソース、予測、エラー、...を管理する
- DB2インタフェース
(表、ビュー、...の) DDLを管理する

容易なカスタマイズ

- ISPFインタフェースはユーザ要求で簡単に調整できます
パッケージ・テンプレート
移動(プロモート)後に自動分析/構築(ビルド)
- マクロ言語で、いくつかの機能追加
プロモート後にDB2にバインド
制御、制限
新機能 (データベース・ツリー、...)