System Maintenance Aid

SMA Concepts

Version 2.1.2

November 2016

This document applies to System Maintenance Aid Version 2.1.2.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Document ID: SMA-CONCEPTS-212-20161110

Table of Contents

1 SMA Concepts

This section describes the basic concepts of SMA.

Note: In this section, the term “tape” represents any installation media (e.g. tape and CD-ROM) supported by Software AG. The information provided here applies to the use of any of these media.

Purpose

Software AG supplies a large and increasing number of products which are constantly being enhanced. This results in a steady flow of new products, new product versions and product corrections to Software AG customers.

These new products and product versions are sent to customers on installation media such as tapes and CDs.

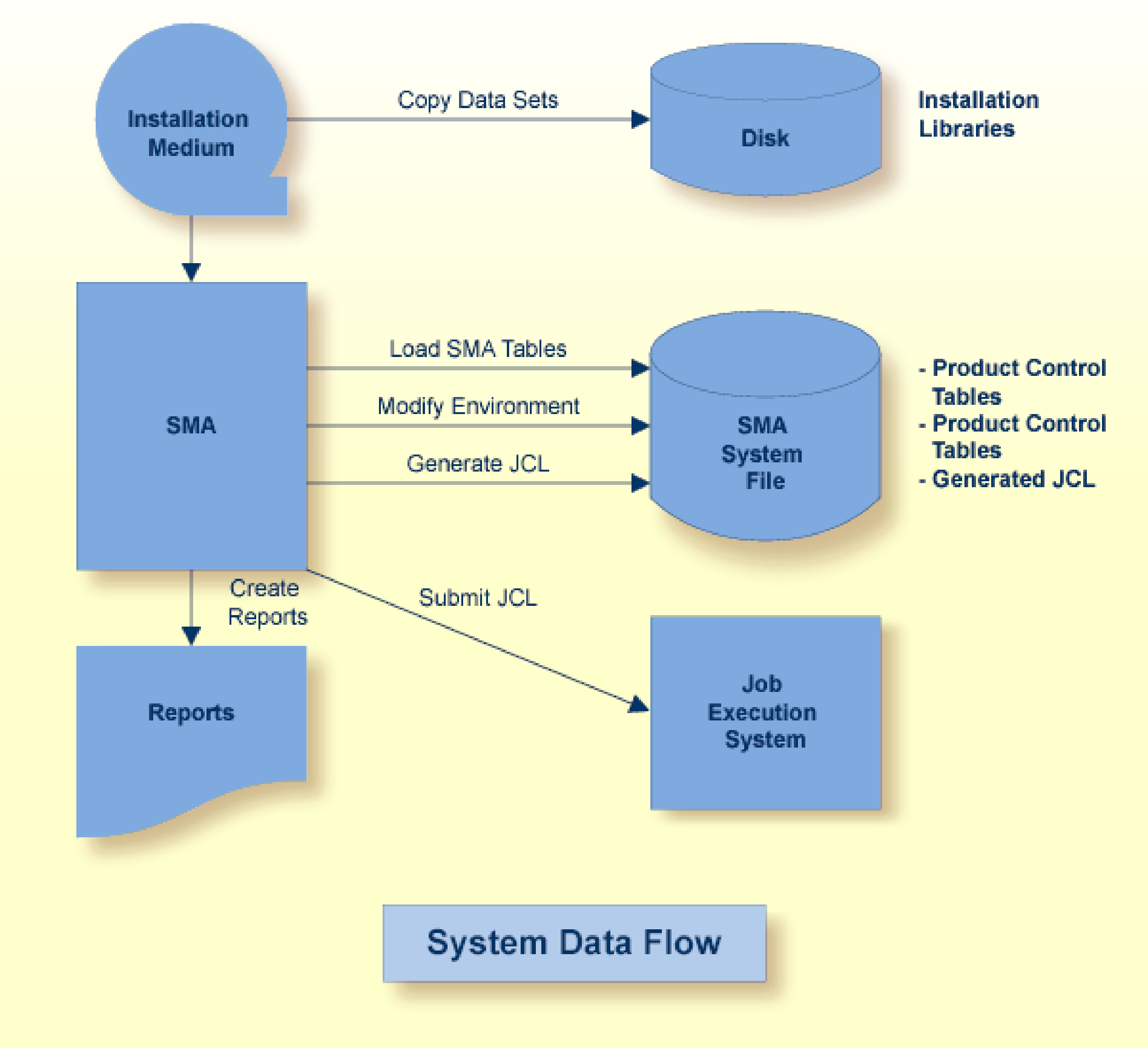
Whenever an installation medium is received, the system administrator must do the following:

- evaluate the contents of the installation medium and
- determine the priorities for the installation of the new product or new product version
- define the environment where the contents should be installed for testing, production or other purposes

These activities are not only required when an installation medium is received. They are also necessary when:

- a new product must be integrated with other Software AG products
- product corrections received from Software AG must be applied

Software AG's *System Maintenance Aid (SMA)* is the tool which can be used to perform the above activities in various operating environments.



Goals

SMA is designed to meet the following requirements:

- provide a uniform mechanism for product installation and maintenance activities for all Software AG products in all customer environments
- provide for easy and flexible incorporation of new products and product versions into the customer's environment
- provide an open system; the Software AG product delivery files remain normal files of the respective operating system, the installation operations are not performed by SMA directly but

rather using generated JCL which can be viewed and modified by the customer as necessary prior to being used to perform the installation

- provide extensive reporting facilities
  - descriptions of product installations; SMA prints all available information about how products are to be installed
  - documentation of the existing installation; what versions of which products have been installed, under what names, the library names used, when the installation was performed

Environment

An important element within the SMA concept is the *User Environment*.

- User Environment Requirements
- SMA User Environment

User Environment Requirements

Most user environments require the usage of multiple Adabas databases and/or multiple Natural sessions. This is usually necessary to accomodate the different requirements imposed upon an Adabas/Natural system. The most important requirements are:

- the system administrator has to test new products, new product versions and new installation options
- application developers need to test data and application programs in parallel to production versions
- end users need reliable and stable application programs and system installations

Each type of requirement is usually serviced by its own specific environment.

Typically, a commercial computer system has environments for the following purposes:

- production
- application development
- system test

There are many different ways of establishing a test and a production environment with Adabas and Natural. Usually, there is one database per environment, each with one Natural system file. However, it is also possible to have several system files representing different environments within one database.

It is also possible for the same Natural system files from different TP systems (for example, TSO and CICS) to use the same database. This is considered to be one environment.

SMA User Environment

An SMA user environment *is represented by a Natural system file*. All user-provided information which describes the installed products and operating environment is stored and used by SMA to control the installation and maintenance activities for the environment.

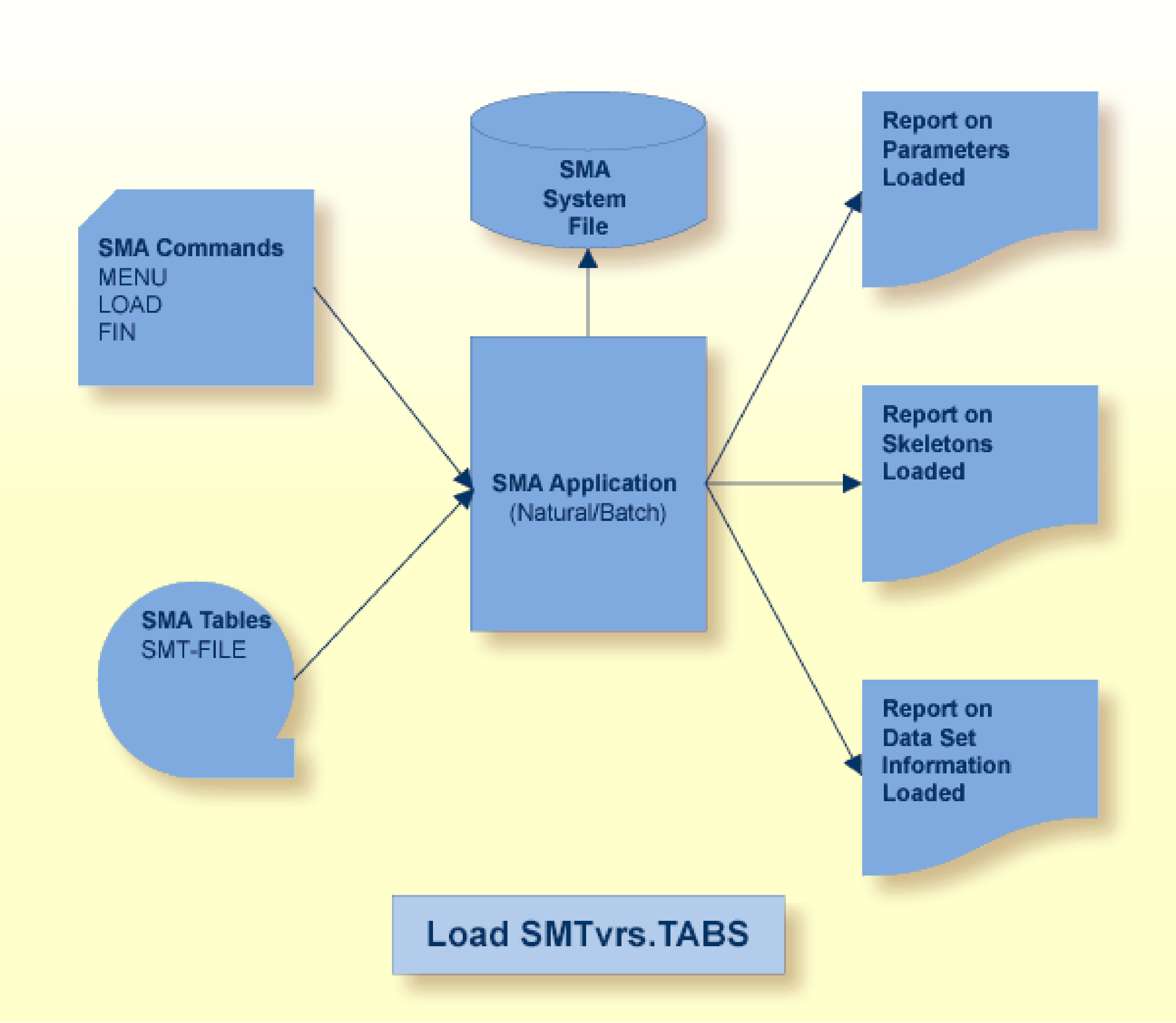
SMA allows the creation of any number of environments.

Product Delivery and Installation

Product Delivery

A product delivered by Software AG consists of a sequence of data sets on a product installation medium (e.g., tape or CD-ROM) as well as the product documentation.

The installation medium contains the Software AG product data sets which are normal files of the corresponding operating system. The first data set on the medium describes the contents of the medium and how it can be processed. The default name is SMT*vrs*.TABS.



An installation medium contains the following types of data sets (the most frequently used data sets are listed below):

- libraries with members containing Assembler source
- libraries with members containing object code
- libraries with members containing load modules
- input data for loading or updating Natural applications
- input data for defining and loading database files

Product Installation

A product installation consists of a sequence of operations using these data sets, as well as applying changes or inserting entries in other software systems.

The following operations are performed using the product data sets:

- allocating/cataloging data sets
- copying data sets or members of libraries
- updating source members, either manually or by merging
- assembling source members to produce object members
- linking object members to produce executable modules
- invoking Software AG utilities (INPL, ADALOD, etc.)
- special functions such as CICS precompiler
- changing the contents of executable modules at specified addresses (Zaps)

New versions or releases of Software AG products usually require a re-installation of the product. Error corrections are supplied as modifications.

A modification can be any of the following (currently only the first two are supported with SMA):

- updates as input for the Natural INPL utility
- changes in executable members (Zaps)
- replacement of an executable member
- replacement of a source member

Installation Control Tables

SMA uses data and programs as described below:

- SMA data. All information for product installation and maintenance is contained in SMA control tables.
- SMA programs. Programs for input and update of control tables, as well as JCL generators which use these tables to produce commands for the operating system.

The most important SMA tables are:

- Environment Description Table

Product Installation Table

The installation of a product under different operating systems is very similar. For example, installation of Adabas usually requires the following:

- reserve disk space
- load the program library
- format the database
- define the database
- start the nucleus

For each product, such a sequence is represented in SMA by a sequence of entries in the *Product Installation Table*. Each entry in this table consists of:

- the name of a function (COPY, ASSEM, LINK, etc.)
- the parameters for the function (for example, which data sets are to be copied)

Each of these entries represents a single step or invocation of a program.

JCL Skeletons Table

For each function in each operating system, there is a *JCL Skeleton* which contains the command statements for the operating system to execute the function. Such a skeleton consists of JCL lines containing parameters. A skeleton is identified by a function name and the name of the operating system.

The following is a sample JCL Skeleton for performing a link edit function under an IBM z/OS operating system. The skeleton parameters are enclosed within the character (#).

```
Op-Sys  OS/MVS    Job-Nr I080  Step-Nr 0020  Skeleton LINK

//LKD#S-SEQ# EXEC PGM=#V-LKED#,
//    PARM=('REUS,XREF,MAP,LET,LIST,NCAL',
//   'SIZE=(#V-LKED-SIZE-1#K,#V-LKED-SIZE-2#K)'),
//    COND=(4,LT)
//SYSLMOD   DD DSN=#P-SYSLMOD#,
//             DISP=SHR
//SYSLIN    DD DDNAME=SYSIN
//SYSUT1    DD UNIT=#V-TEMPUNIT#,SPACE=(1700,(500,100))
//SYSPRINT  DD SYSOUT=#V-SYSOUT#,
//             DCB=(RECFM=FB,LRECL=121,BLKSIZE=1210)
#I-P-LIB1#
//SYSIN     DD DSN=#P-SYSIN#,DISP=SHR
```

```
//SYSIN     DD *
#I-P-PHASE#
//*
```

Environment Description Table

An *Environment Descriptor Table* contains additional information which is required to expand the data in the Product Installation Table in order to create an executable command sequence, and also to select the correct JCL skeleton. This information is customer- and installation-specific but not product-specific.

Each Environment Description Table contains the following information for one environment:

- products installed in the environment
- the specific parameter values which are to be inserted into the JCL skeletons during JCL generation

JCL Generation

Creating Executable Control Statements

The process of JCL generation is as follows:

- The SMA user marks the products to be installed
- Using the Product Installation Table of these products, SMA
  - takes the JCL skeletons as indicated in the Product Installation Table
  - and replaces the parameters in the skeletons with parameter values taken from the user's environment description

The JCL steps are not necessarily generated in product order, but rather in the order of the job names and steps as contained in the Product Installation Table. Therefore, the generated jobs will contain groups of similar functions; for example, all database load operations or all assemblies.

Interrelationship between Products

Certain Software AG products require other products as prerequisites; for example, the product Predict requires the product Natural. These dependencies are checked within SMA.

In addition, there may be dependencies in the installation processes for different products as shown by the following examples:

- assembly and link: a module is assembled as part of the installation of one product (for example, Natural CICS Interface), and the resulting object module must be included in the link edit operation of a second product (for example, Natural).
- parameter modules: one product (for example, Con-nect) requires certain parameter settings in source modules of other products (for example, Natural).

These dependencies are resolved based on conditions contained in the JCL skeletons. These conditions can be based on the products installed, on batch mode or online mode execution, or the values of parameters.

By applying these conditions, it is possible to include or exclude statements in the parameter modules or in linkage editor input.

Libraries

Library Requirements

The JCL generated by SMA assumes that data sets and libraries are to be used in a certain way. This usage must be based on a sound library concept. Some requirements for a good library concept are:

- the exact version and status of an executable member and of all members which have been used to create the executable member must be traceable
- no action concerning one version of a product (for example, copy from tape to disk, or link edit) is allowed to affect another product version
- it must be possible to work on different versions of a product concurrently
- customer-specific modules must not be overwritten when new product data sets are loaded
- job control statements must not contain version-dependent names of libraries or modules

The following conventions implement the above requirements. As far as possible, these rules are independent of the operating system.

Library Organization and Contents

The data sets used for installation at the customer site must be organized in the same way as the delivery libraries provided by Software AG; that is, there are load and source libraries for each version of each Software AG product.

In addition, there are work libraries which contain the following:

- SMA.LOAD:

  All results of the link steps which produce executable modules (phases) and of assembly/link steps of those source programs which have been adapted for the environment.

- SMA.SOURCE:

  Source members generated by SMA.

Llibrary names can be specified using the following parameters:

Operating System	Parameter
z/OS	DSN-SMALOAD DSN-SMASRCE
z/VSE	USRLIB
BS2000/OSD	JOBLIB

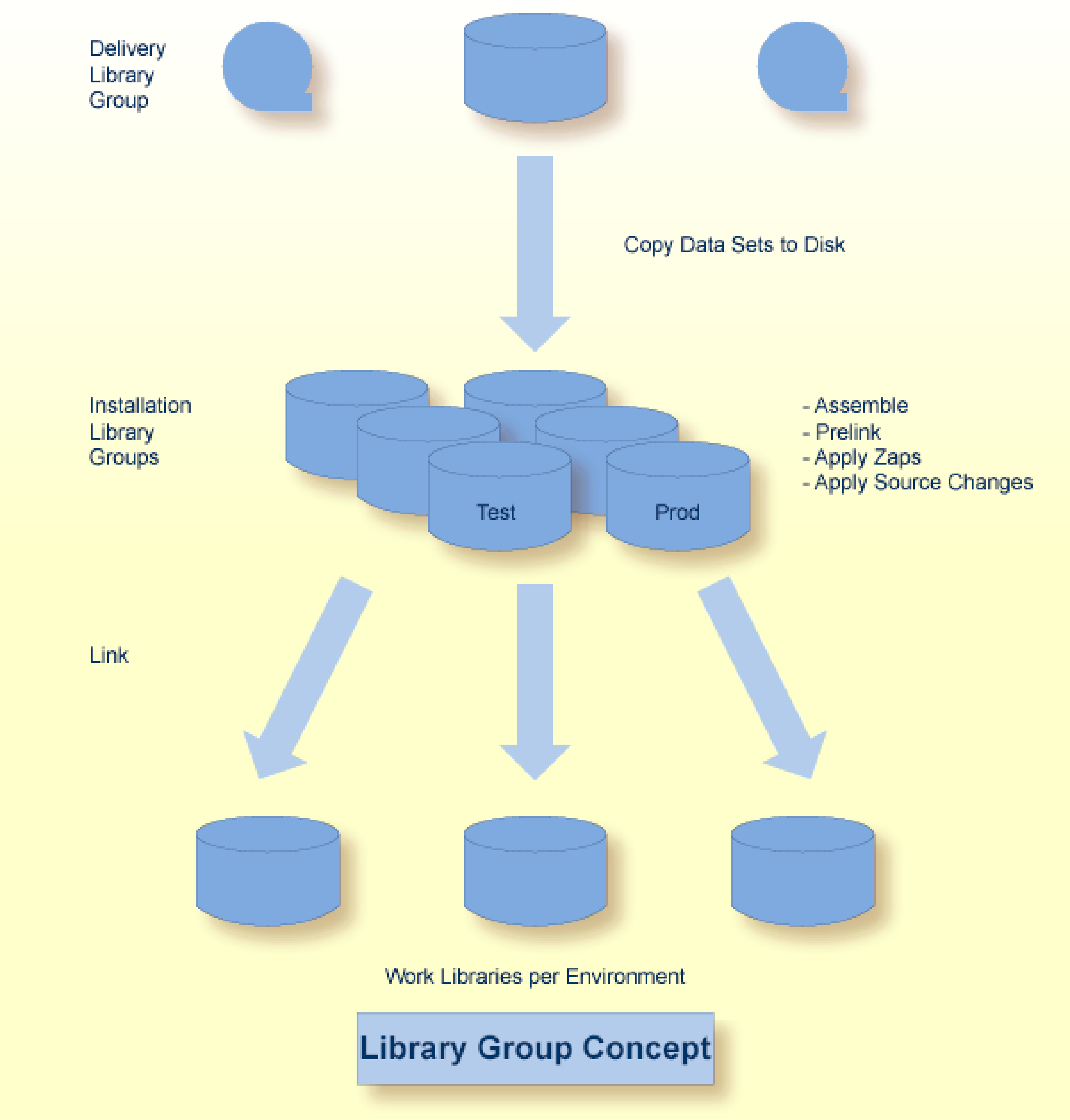
There is one set of work libraries for each environment. There can be one or more installation libraries for an operating system (see Library Groups below). For more information on library usage for a particular operation system, see the following sections:

- Using SMA under z/OS
- Using SMA under z/VSE
- Using SMA under BS2000/OSD

Library Groups

The installation libraries are changed by the installation process, and also by corrective actions. A good library concept must allow for applying such changes selectively for different environments; one reason being the requirement for testing of corrections. SMA does not require a separate copy of each library for each environment, because the resulting number of library copies might be considered unacceptable.

SMA provides the concept of *Library Groups*. Library groups provide the flexibility to adapt the library concept to individual requirements.



There is always one default library group, the delivery group, which is specified in the default environment and is used to keep track of the delivery files. The default name of the delivery group is SAGLIB for z/OS and z/VSE and $SAGLIB for BS2000/OSD.

The SMA user has the following options for introducing additional library groups:

- Only One Group

In situations where there is little disk space, or where minimal control of the status of the libraries is required, the same library group may be used in the default environment and in all installation environments.

In this setup, the SMA user must carefully control the sequence of maintenance and installation actions.

- One Group for Test and One for Production

  One library group is used in the installation environment for test and development, and one in the production environment. This is the recommended setup.

- One Group for each Environment

  Using a different library group for each environment results in independent environments, and provides the SMA user with maximum control. The disadvantage of this method is the need for creating a possibly large number of installation libraries, depending on the number of environments and different Software AG products in use.

Llibrary names can be specified using the `LIB-GROUP` parameter.

Library Installation and Maintenance

Copying Libraries from Tape to Disk

The SMA Tape Management function can be used to select tape data sets which are to be copied to disk.

Whenever JCL generation for product installation is requested, all library data sets will be checked to determine whether or not they are already on disk for the group which is used in the chosen environment. If not, the appropriate copy steps are generated to be executed before the installation steps.

Assemblies and Prelinks

Assemblies take their input from one of the source libraries of the installation pool. The results of these assemblies are stored in the environment work load library.

Linking Executable Modules

The link steps which produce the executable modules place their results in the environment work load library. The names of the executable members contain the version number (for example, NCI414T1).

Zaps

Zaps are applied to the members in the group installation libraries. SMA keeps track of all Zap applications, and knows which executable modules in which environments must be re-linked after any component member has been changed.