

Concepts of System Maintenance Aid

This section covers the following topics:

- Purpose of Software AG System Maintenance Aid
 - Goals of System Maintenance Aid
 - Environment
 - Products and Product Installations
 - Installation Control Tables
 - JCL Generation
 - Library Concept Used by SMA
 - JCL Generation in Different Operating Systems
-

Purpose of Software AG System Maintenance Aid

Software AG supplies a large and ever-increasing number of products, and these products are constantly being enhanced. This productivity 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 "System Maintenance Tapes".

Each time a customer receives a "System Maintenance Tape", the system administrator must do the following:

- Evaluate the contents of the tape and determine the priorities for the installation of the new versions
- Install one or several new versions for different tests
- Install one or several new versions for production.

Normally, this cycle extends over a period of time, and usually the installation steps are repeated for test and production. This type of work is not only required when "System Maintenance Tapes" are to be processed. It also arises when:

- a new product has been ordered from Software AG and has to be integrated with other Software AG products;
- product corrections have been received from Software AG which must be applied for test and for production.

Software AG's System Maintenance Aid is designed to support this type of work for an increasing number of products which are usable in different operating environments.

SMA also supports the installation of Software AG products with new Software AG customers. SMA is the tool for the customer's system administrator as well as Software AG's customer support personnel.

Goals of System Maintenance Aid

System Maintenance Aid is designed to meet the following requirements:

- A uniform mechanism for the product installation and maintenance activities for all Software AG products in all customer environments.
- A high degree of flexibility: Each new product or version arriving at a customer's site can be incorporated easily into the customer's SMA installation.
- 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, and the generated JCL can be viewed and modified by the customer if the need arises.
- Extensive reporting facilities:
 - Descriptions of product installations: SMA prints out all its "knowledge" about how products are to be installed.
 - Documentation of the existing installation: What versions of which products have been installed under what names; what are the library names used; when was this done; etc.

Environment

The central subject within the concepts of SMA is the environment.

Most users of Adabas and Natural will operate several databases or several Naturals. This is necessary to separate different requirements imposed upon an installed Adabas/Natural system. The most important requirements are:

- The system administrator has to test and inspect new products, new product versions and new installation options.
- Application developers need test data as well as test and development versions of application programs in parallel to production versions.
- Users need reliable and stable application programs and system installations.

Each group of requirements should be serviced by a different environment.

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

- Production
- Application development
- System test

Note that 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, but it is also possible to have several system files within one database. These would also be considered to be distinct environments.

It is also possible to use the same database with the same Natural system files from different TP-systems (e.g., TSO and CICS). These are considered to be one environment.

An SMA environment is, thus, represented by a Natural system file. Each environment has a number of products installed in it.

SMA allows the creation of any number of environments. All information entered by the user describing the installed products and operating environment are stored within SMA in specific environments. Therefore, products are always installed within a selected environment.

Products and Product Installations

A product delivered by Software AG consists of a sequence of datasets on tape and documentation.

There are the following types of datasets:

- 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.

An SMA tape is a product delivery tape sent by Software AG to the customer. It contains the Software AG product datasets which are "normal" files of the corresponding operating system. The first dataset on the tape describes the contents of the tape and how it can be handled.

Installation Operations

An installation consists of a sequence of operations using these datasets and of changes or entries in other software systems.

There are the following operations using the product datasets:

- Allocation / cataloging of datasets
- Copying datasets or members of libraries
- Updating source members, either manually or by "merging"
- Assembly of source members to produce object members
- Linking object members to produce executable modules
- Invoking Software AG utilities (INPL, ADALOD, etc.)
- Special functions like CICS precompiler
- Changing the contents of executable modules at specified addresses (ZAPs).

This list of functions represents nearly all operations which occur during product installation, but this list is neither fixed nor predefined.

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:

- Changes in executable members (ZAPs)
- Replacement of an executable member
- Replacement of a source member
- Updates as input for the Natural INPL utility

Installation Control Tables

SMA consists of programs and data:

- All information on products, their installation and their maintenance is contained in database tables and control data.
- The SMA programs consist of facilities for input and update of these control tables and of generators which use these tables to produce commands for the operating systems.

The most important tables are described in the following sections:

- Product Installation Table
- JCL Skeletons Table
- Environment Description Tables

Product Installation Table

An analysis of the installation of a product under different operating systems shows that the basic process is very similar in all systems. For example, installation of Adabas can be described in the following terms:

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

For each product, such a sequence of actions is represented 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" of this function (e.g., which datasets 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 "skeleton" containing the command statements for the operating system to call and execute this function. Such a skeleton consists of JCL lines containing formal parameters, and is identified by a function name and the name of the operating system.

The following example shows a possible skeleton for link-edit under an IBM/OS operating system. The formal parameters are enclosed between pairs of number signs (#).

```
//S0001 EXEC PGM=IEWL,
// PARM='LIST,LET,NCAL,REUSE,RENT'
//SYSUT1 DD UNIT=#V-TEMPUNIT#,
//      DISP=(NEW,DELETE,DELETE),
//      SPACE=(TRK,(#V-TEMPSPACE#,10))
//IN1 DD DSN=#D-DSN-DISK(NAT234.LOAD)#,
//      DISP=SHR
//SYSLIN DD DDNAME=SYSIN
//SYSIN DD *
```

Environment Description Tables

Additional information is required to expand the data of the Product Installation Table to an executable command sequence and to select the correct JCL skeleton. This information is not product-specific, but customer- and installation-specific. Therefore, tables describing the customer site and environment are required.

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

- products installed in the environment;
- the specific parameter values which have to be inserted in the JCL skeletons during JCL generation.

JCL Generation

The following topics are covered below:

- Creating Executable Control Statements
- Interrelationship between Products

Creating Executable Control Statements

The process of JCL generation is as follows:

- The user marks the products to be installed.
- SMA works through the Product Installation Table of these products,
- takes the JCL skeletons as indicated in the entries of 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 the order of products, but in the order given by the names of the jobs and steps in the Product Installation Table. Therefore, the 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, Predict requires Natural. These dependencies are checked within SMA.

In addition, there are dependencies in the installation processes of different products. These dependencies occur in two situations:

- Assembly and Link: A module is assembled as part of the installation of one product (e.g., Natural CICS Interface), and the resulting object module must be included in the link-edit operation of a second product (e.g., Natural).
- Parameter Modules: One product (e.g., Con-nect) requires certain parameter settings in source modules of other products (e.g., Natural).

These relationships are treated in the following way: The lines in the JCL skeletons contain conditions which are based on the products installed, on batch or online execution, or on 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.

Library Concept Used by SMA

The following topics are covered below:

- Requirements for a Library Concept
- Installation Libraries
- Library Groups
- Installation and Maintenance Operations

Requirements for a Library Concept

The JCL generated by SMA assumes datasets and libraries 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 (e.g., copy from tape to disk, or link-edit) is allowed to affect another version.
- It must be possible to work on different versions of a product concurrently.
- Customer-specific modules must not be overwritten when new product datasets are loaded.
- Job control statements supplied by the user 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.

Installation Libraries

The datasets used for installation at the customer site must be organized in the same way as the delivery libraries of Software AG, i.e., there are load and source libraries for each version of each Software AG product.

In addition, there are work libraries containing the following objects:

- **SMA.LOAD:**
All results of the link-steps producing executable modules ("phases") and of assembly/link-steps of those source programs which have been adapted per environment.
- **SMA.SOURCE:**
Source members generated by SMA.

You can specify the library names using the following parameters:

OS/MVS	DSN-SMALOAD DSN-SMASRCE
VSE/SP	USRLIB
BS2000	JOBLIB

The work libraries exist once per environment, whereas the installation libraries exist once or several times per system (see Library Groups below).

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 test 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.

This is achieved by the concept of "Library groups". Library groups provide the user with the flexibility to adapt the library concept to individual requirements.

There is always one default library group, the "delivery group". This library group is specified in the default environment, and it is used to keep track of the delivery files. The default name of the delivery group is SAGLIB.

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

Only One Group

In situations where there is little disk space, or when minor 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 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 environments for test and development, and one in the production environment(s). This is the recommended setup.

One Group for each Environment

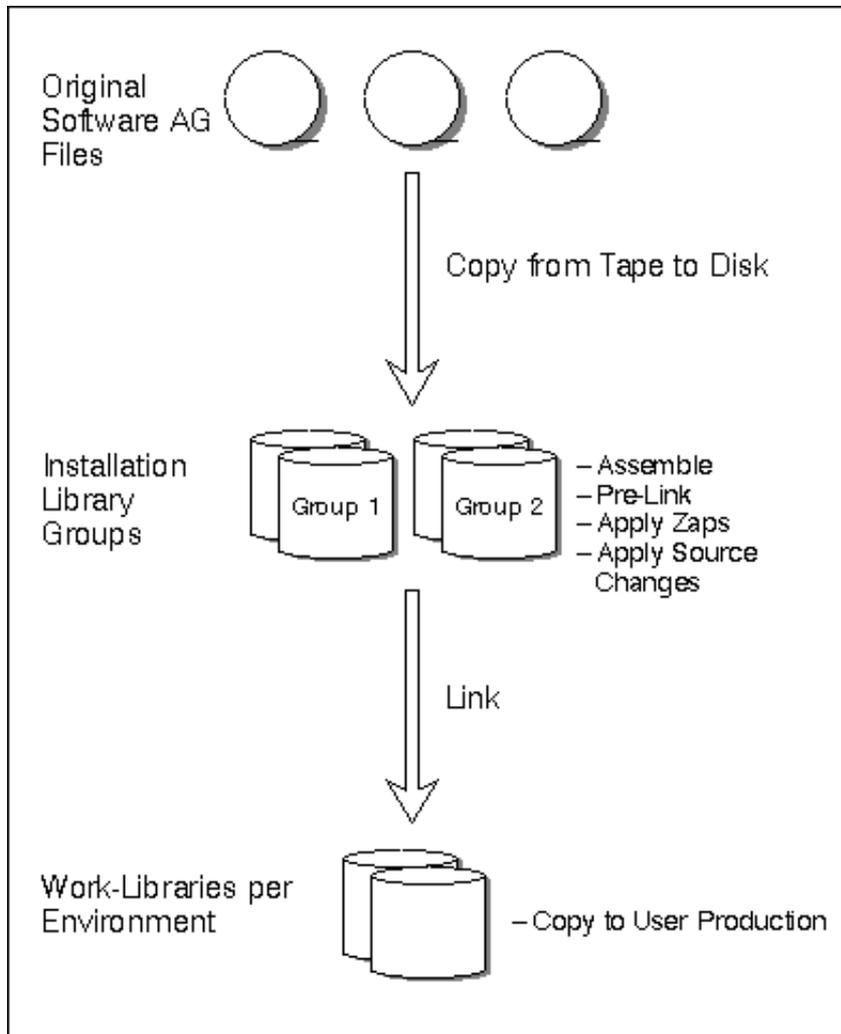
Using a different library group for each environment results in independent environments, and provides the user with maximum control. The disadvantage of this method is the need for creating a possibly large number of installation libraries, depending on the number of environments and different Software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

You can specify the library names using the following parameters:

OS/MVS	LIB-GROUP
VSE/SP	LIB-GROUP
BS2000	LIB-GROUP

Installation and Maintenance Operations

Installation and maintenance operations of Software AG products use these libraries. The figure below demonstrates the library concept, and operations on it.



Copying Libraries from Tape to Disk

In the tape management part of SMA, the user may select tape datasets to be copied to disk.

Whenever JCL generation for product installation is requested by the user, all library datasets will be checked to determine whether 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 workload 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 (e.g., NCI216T1).

ZAPs and Source Changes

ZAPs and source changes 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 relinked after any component member has been changed.

JCL Generation in Different Operating Systems

Most of SMA's concepts are valid for all operating systems under which SMA can be used. The following sections describe some SMA concepts which are specific for the different operating systems. This applies in particular to executing the JCL generator and storing the generated JCL.

- BS2000
- OS/MVS and VSE/SP

BS2000

The "JCL" generated for BS2000 consists of the following parts:

- Environment-specific procedures are generated for common tasks, like loading an Adabas file, assembling and linking, etc. These procedures are generated according to the parameters given for a specific environment (and thus belonging to this environment), but they are used in the installation jobs with different parameters.
- Most of the installation jobs consist solely of calls to these procedures.
The user may apply these procedures also in tasks other than the SMA-generated installation jobs.
The procedures and jobs are stored in an LMS library or as separate files as desired by the user.
- Source datasets for different purposes are also created.

Main Topics

The following topics should be considered:

- JCL is separated into jobs, "central" procedures and data files.
- Installation from disk and from tape.
- LMS: LMS is required for delivery libraries as well as for installation JCL libraries and job output libraries.
- Library separation: Delivered libraries and libraries used during the installation are separated. Modules which are created during installation are not located within the delivery libraries.
- Spin-off routine: all jobs stop immediately if an error occurs. All jobs have controlled normal end (normend) and abnormal end (abend) exits. All jobs and all job steps write an oneline message about their normend/abend to the file L.REPORT.
- Restart routine: If an installation job ended abnormally or was canceled, it can be restarted. The job will then start processing with the interrupted job step. (Job variables are used for this feature.)
- Automated Installation: A control job is generated which can be used to start and control all installation jobs generated.

Additional Information

- All members in JOBLIB have information headers.
- The skeleton #READ-ME.... contains special installation information concerning the product.
- Also refer to:
 - #NEWS-FOR.SMA
 - #READ-ME.SMA
 - #READ-ME.@IS
 - #READ-ME.@SD

OS/MVS and VSE/SP

OS or VSE users may call the SMA JCL generator either in batch or within the SMA dialog. Both methods have their specific advantages:

- JCL generation in dialog is the method SMA has been designed for: Jobs are created according to the specifications given by the user, submitted from within the SMA dialog and deleted after successful execution.
- The user may save generated jobs within SMA, and may unload stored jobs via batch execution of the UNLOAD JOBS command of SMA.
- When called in batch, the SMA JCL generator writes the JCL generated into its system file as well as into a work file 2. The work file contains separators in IEBUPDTE or LIBR format between the different jobs to allow for loading this output into libraries. See the job JCLGEN from the SMAnnn.SRCE library as an example.
- The generated jobs are created using the same set of job names with each JCL generation. Users who wish to keep generated jobs must use a JCL library and a naming convention for their own jobs.