INSTALLATION

Predict Version 3.4 or above is a prerequisite for the installation of Adabas Native SQL. Please check which version of Predict is installed.

The Adabas files and fields that will be used by Adabas Native SQL application programs must be defined in the data dictionary.

This chapter covers the following topics:

- Positioning the Medium

- Copying to a VM/CMS Disk

- Loading Error Messages

- Modifying the Sample EXEC Procedures

- Modifying the Global Parameters

- Modifying the DDCARD Parameters

- Testing Adabas Native SQL

- FILEDEF and DATADEF Names

Positioning the Medium

Adabas Native SQL consists of two datasets. To determine how to correctly position the tape in the next two sections *Loading Files from Magnetic Tape* and *Loading of Error Messages* please refer to the *Software AG Product Delivery Report*.

Copying to a VM/CMS Disk

To position the tape for the TAPE LOAD command, calculate the number of tape marks as follows:

- If the sequence number of SQL23n.TAPE - as shown by the Software AG Product Delivery Report - is n, you must position over 3n-2 tape marks (that is, FSF 1 for the first dataset, FSF 4 for the second, etc.).

- Access the disk that is to contain the Adabas Native SQL installation files as minidisk.

- Ask the system operator to attach a medium drive to your virtual machine at the address X'181' and mount the Adabas Native SQL installation medium.

When the medium has been attached, enter the VM/CMS command:

```
TAPE REW
```

Position the medium by entering the VM/CMS command:

```
TAPE FSF n
```

where n is calculated as above (3*n*-2).

Load the Adabas Native SQL/VM/CMS installation material with the VM/CMS command:

```
TAPE LOAD * * <minidisk>
```

You may wish to keep the medium drive attached to your virtual machine, because the medium is still needed in the installation procedure.

Loading Error Messages

The other dataset of the SQL*nnn* medium is an error messages dataset for the Adabas Native SQL preprocessor. The dataset name is SQL*nnn*.ERRN. The following commands can be used to load the error messages:

```
TAPE REW
FILEDEF CMWKF02 SLn VOLID SQLnnn
                      (LRECL 4624 BLCKSIZE 4628 RECFM VB)
FILEDEF CMPRT01 DISK SQLnnn INPL A
NAT3vs MAINPR=1,STACK=(LOGON SYSERR;ERRLODUS;FIN)
TAPE RUN
```

Where *n* represents the label defined in the Software AG Product Delivery Report and *vs* represents the Natural version and SM level.

Modifying the Sample EXEC Procedures

Modify (XEDIT) the supplied EXECs ADASQLC, ADASQLF, ADASQLP and ADAGO as required for execution of the sample programs. The following specifications in the EXECs may be adapted to suit the system requirements.

- It may be necessary to provide the LINK and ACC statements necessary to acquire access to the Adabas virtual machine and to the Adabas text library.

- It may be necessary to provide the LINK and ACC statements necessary to acquire access to the COBOL system libraries for the COBOL compiler.

- The FILEDEF cards may be modified to point to different minidisks.

- The COBOL compile options (parameters) may be modified.

- In the ADAGO.EXEC, the LOAD statement may need to be modified depending on which global parameters are used. The module FINDCID is included only if the global parameter OPTIONS DYNAMCID is specified. The module PRTFLOW is included if the the global parameter MODE FLOW is specified. These modules will work in both COBOL and PL/I application programs.

In PL/I only environments, overhead can be further reduced by performing the following:

1. Compile routines RESPPL1, PRTRACP and PRTFLOP.

2. Code global parameter `ABORT RESPPLI PLI` in ADAGLOB ADASQL.

3. In the LOAD statement, replace RESPINT, PRTRACE and PRTFLOW with RESPPL1, PRTRACP and PRTFLOP.

The EXEC as supplied loads the standard Adabas interface module Adabas for the application program. If a different module is used, the EXEC must be modified accordingly.

Modifying the Global Parameters

The global parameters for Adabas Native SQL must also be modified by editing (XEDIT) ADAGLOB ADASQL fm.

Note that the DDFILE parameter has been replaced by the SYSFILE parameter. Adjust the FDIC and FNAT parameters accordingly.

If the error message texts are stored in a file other than the data dictionary file, the parameter `FNAT` should be used to specify the number of this file. The error message texts are typically stored in the Natural system file (FNAT).

See chapter *Global Parameters* in the *Adabas Native SQL Reference Manual* for further information.

Typical global parameter definitions:

```
SYSFILE FDIC=(DBID,FNR) FNAT=(DBID,FNR).
OPTIONS SOFT=ON NONDE=Y PREFIX=''.
XREF ON.
LANG COBOL.
```

Modifying the DDCARD Parameters

The Adabas runtime parameters must also be modified by editing (XEDIT) DDCARD ADASQL fm. See the *Adabas CMS Installation Manual* for information regarding parameter usage.

Testing Adabas Native SQL

Before testing the examples, check that the files EMPLOYEES and VEHICLES are loaded, and that they are documented in the Predict data dictionary. If the file definitions are not yet present in the data dictionary, they can be loaded from the file PRD*nnn*.DEMO, which can be found on the Predict installation medium. The notation *nnn* represents the Version, Release and SM level of Predict at your site. If you wish to use Adabas Native SQL to preprocess FORTRAN programs, ensure that the necessary language synonyms are defined in the data dictionary. See for example Appendix B in the *Adabas Native SQL Reference Manual*.

It may be necessary to increase the storage capacity of the virtual machine that will execute the Adabas Native SQL preprocessor. To inquire on the amount of storage the virtual machine has, execute the following z/VM command:

```
Q STOR
```

If the response indicates that there is less than 2048k available, enter the following z/VM commands:

```
DEFINE STOR 2M
IPL CMS
```

The EXEC ADASQLC, ADASQLF or ADASQLP executes the following:

1. Prompt for the file name, file type and mode. File name will be the name of the example to be compiled, file type must always be source, and mode is the letter corresponding to the minidisk where the source of the example is contained (i.e. CEX1 SOURCE C).

2. Preprocess the program with Adabas Native SQL.

3. Compile the preprocessed program (COBOL, FORTRAN, PL/I).

4. The following modules may be called by the user program at runtime:

Ada	COBOL	FORTRAN	PL/I	Description
RESPF (F)	RESPINT (C)	RESPF (F)	RESPINT (C,P)	Error handling routine.
PRTRAC (F)	PRTRACE (C)	PRTRAC (F)	PRTRACE (C,P)	Runtime trace routine.Activated by the error handling routine if `MODE TRACE` is set.
PRTFLO (F)	PRTFLOW (C)	PRTFLO (F)	PRTFLOW (C,P)	Runtime Adabas Native SQL statement printing. Activated if `MODE FLOW` is set.
FCID (A)	FINDCID (A)	FCID (A)	FINDCID (A)	Generates dynamic Adabas command IDs. Activated if `OPTION DYNAMCID` is set.
PRPABEND (A)	PRPABEND (A)	--	PRPABEND (A)	Terminates program execution. Called by the error handling routine.
SQFRDATE (C)	SQFRDATE (C)	SQFRDATE (C)	SQFRDATE (C)	Routine which converts format D number to numeric date.
SQFRTIME (C)	SQFRTIME (C)	SQFRTIME (C)	SQFRTIME (C)	Routine which converts format T number to numeric date and numeric time.
SQTODATE (C)	SQTODATE (C)	SQTODATE (C)	SQTODATE (C)	Routine which converts numeric date to format D number.
SQTOTIME (C)	SQTOTIME (C)	SQTOTIME (C)	SQTOTIME (C)	Routine which converts numeric date and numeric time to format T number.

Notes:

1. The FORTRAN error handling routine RESPF always returns condition code 8 when the job step terminates. The other error handling routines return the Adabas response code.
2. The characters in parentheses indicate the language in which the routines are written

(Assembler, COBOL, FORTRAN or PL/I). Only the Assembler and COBOL modules are supplied in object form; if you wish to use FORTRAN or PL/I modules, you must compile the source modules.

3. The COBOL object modules of the routines RESPINT, PRTRACE and PRTFLOW supplied by Software AG can be linked with PL/I object modules to produce an executable load module. If you wish to use the PL/I versions of these routines, you must rename them: RESPPL1 - RESPINT, PRTRACP - PRTRACE, PRTFLOP - PRTFLOW, and then compile them, replacing the Software AG-supplied COBOL object modules by the PL/I object modules. The global parameter ABORT PLI should be coded. The procedure as supplied links the standard Adabas interface module ADAUSER to the application program. If a different module is used, the procedure must be modified accordingly.

5. Upon completion, notification is made as to the location of files created during preprocess and compilation. You should review these files for error and/or warning messages before continuing.

6. Execute the ADAGO EXEC.

If the program is to be passed through other preprocessors in addition to Adabas Native SQL, the additional preprocessor(s) should be included in the EXEC following the Adabas Native SQL step and the preceding compilation.

FILEDEF and DATADEF Names

Adabas Native SQL refers to datasets using the following names:

FILEDEF	Description
ADAIN	The source program to be preprocessed.
ADAOUT	The output of the preprocessor. Embedded Adabas Native SQL statements in the source program appear in this dataset as comments and are followed by the generated COBOL or PL/I code.
ADAGLOB	The global parameter definitions.
ADAMES	Adabas Native SQL writes messages at preprocessing time to this dataset.
SYSUT1	A temporary work file.
SYSDBOUT	Error messages are written to this dataset.
SYSOUT	Adabas Native SQL writes messages to this dataset if it abends.

At run time, the application program generated by Adabas Native SQL refers to datasets using the following names (in addition to any other user-defined datasets):

FILEDEF	Description
SYSOUT	If the FLOW facility is used, the flow-tracing output is written to this dataset.
TSTDMP	If the TRACE facility is used, the trace output is written to this dataset.

With Adabas Version 5, a DATADEF must be issued for the following files:

DATADEF	Description
DDPRINT	Adabas writes messages to this dataset at run time.
DDCARD	Adabas parameters that are used by the ADARUN module at run time.