

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:

- Copying to an z/OS Disk
 - Loading Files from Medium
 - Modifying the Sample JCL Procedures
 - Modifying the Global Parameters
 - Relinking Adabas Native SQL
 - Testing Adabas Native SQL
 - DD-Names
-

Copying to an z/OS Disk

If you are using System Maintenance Aid (SMA), refer to the SMA documentation (included on the current edition of the Natural documentation CD).

If you are not using SMA, follow the instructions below.

This section explains how to:

- Copy data set COPY.JOB from medium to disk.
- Modify this data set to conform with your local naming conventions.

The JCL in this data set is then used to copy all data sets from medium to disk.

If the datasets for more than one product are delivered on the medium, the dataset COPY.JOB contains the JCL to unload the datasets for all delivered products from the medium to your disk.

After that, you will have to perform the individual install procedure for each component.

Step 1: Copy data set COPY.JOB from medium to disk

The data set COPY.JOB (label 2) contains the JCL to unload all other existing data sets from medium to disk. To unload COPY.JOB, use the following sample JCL:

```
//SAGTAPE JOB SAG,CLASS=1,MSGCLASS=X  
//* -----  
//COPY EXEC PGM=IEBGENER
```

```
//SYSUT1 DD DSN=COPY.JOB,
// DISP=(OLD,PASS),
// UNIT=(CASS,,DEFER),
// VOL=(,RETAIN,SER=<Tnnnnn>),
// LABEL=(2,SL)
//SYSUT2 DD DSN=<hilev>.COPY.JOB,
// DISP=(NEW,CATLG,DELETE),
// UNIT=3390,VOL=SER=<vvvvvv>,
// SPACE=(TRK,(1,1),RLSE),
// DCB=*.SYSUT1
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
//
```

Where:

<hilev>	is a valid high level qualifier.
<Tnnnnn>	is the medium number.
<vvvvvv>	is the desired volser.

Step 2: Modify COPY.JOB to conform with your local naming conventions

There are three parameters you have to set before you can submit this job:

- Set HILEV to a valid high level qualifier.
- Set LOCATION to a storage location.
- Set EXPDT to a valid expiration date.

Step 3: Submit COPY.JOB

Submit COPY.JOB to unload all other data sets from the medium to your disk.

Loading Files from Medium

Use a batch Natural job similar to the following to load the error messages into the system file:

```
//NATB EXEC PGM=batch-natural,REGION=2400K,TIME=1400,
// PARM='IM=D,MADIO=0,MAXCL=0,MT=0,AUTO=OFF'
//STEPLIB DD DSN=batch-natural-load-library,DISP=SHR
// DD DSN=adabas-load-library,DISP=SHR
//DDPRINT DD SYSOUT=*
//DDDRUCK DD SYSOUT=*
//CMPRINT DD SYSOUT=*
//MPMDUMP DD DUMMY
//DDKARTE DD DUMMY
//DDCARD DD *
ADARUN DA=dbid,DE=3390,SVC=249,MODE=MULTI
//CMWKF02 DD DSN=SAGLIB.SQLnnn.ERRN,DISP=SHR
//CMSYSIN DD *
LOGON SYSERR
ERRLODUS
```

```
FIN
/*
//*
//
```

This job is supported by SMA, Job I061, Step 3602. Input file `SQLnnn.ERRN` is used from medium.

Modifying the Sample JCL Procedures

Modify (edit) the supplied JCL procedures ADAGO, COBOLGO, FORTGO, PLIGO and GO as required and then execute the example programs. The following specifications in the procedures should be adapted to suit the system environment. The standard values of the procedure parameters are given in parentheses.

- The Adabas load library index (`ADABAS.Vmmm`)
- The Adabas Native SQL load library index (`SQLnnn`)
- The name of the Adabas Native SQL preprocessor module (e.g. `ADASQLC` in job COBOLGO)
- Language-specific libraries (`SYS1.COBLIB`, `SYS2.LINKLIB` for COBOL; `SYS1.VFORTLIB` for FORTRAN; `SYS1.PLIBASE`, `SYS2.PLI.LINKLIB` for PL/I)
- The SYSOUT classes of the output datasets (X)
- The membername of the program to be preprocessed (AEX1, CEX1, FEX1 or PEX1)
- The name of the library that contains the member to be preprocessed (`SQLnnn.SRCE`).

Modifying the Global Parameters

The global parameters for Adabas Native SQL must also be modified.

Enter the number of the Predict data dictionary file in the SYSFILE FDIC parameter.

The error message texts for decoding preprocessor-time errors are typically stored in the Natural system file, specified by the global SYSFILE FNAT parameter.

The error messages for decoding run-time errors are typically stored in the data dictionary file. If this is not the case, use the global parameter ABORT FILE to specify the location (number) of this file.

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

Typical global parameter definitions:

```
SYSFILE FDIC=(1,11) FNAT=(1,8).
LANG COBOL.
XREF ON.
ABORT FILE=8.
```

Relinking Adabas Native SQL

If a different Adabas interface module is to be used instead of the standard module ADAUSER, Adabas Native SQL must be relinked. The procedure LINKSQL can be used to link the module ADASQLA, ADASQLC, ADASQLF or ADASQLP.

The module ADAUSER uses the DD-name DDCARD to refer to the dataset containing ADARUN parameters. If an alternative module is used, this dataset may not be needed.

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*.

The procedure ADAGO, COBOLGO, FORTGO or PLIGO executes the following steps:

1. List the source program;
2. Preprocess the program with Adabas Native SQL;
3. Compile the preprocessed program (Ada, COBOL, FORTRAN or PL/I);
4. Link-edit the program. The modules listed below may be called by the user program at runtime:
5. Execute the program.

If the program is to be passed through other preprocessors in addition to Adabas Native SQL, the corresponding job step should be included in the procedure following the Adabas Native SQL step and preceding compilation.

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

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

DD-Names

Adabas Native SQL refers to datasets using the following DD-names:

DDNAME	Description
ADAIN	The source program to be preprocessed. LRECL=80,RECFM=FB
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. LRECL=80,BLKSIZE=3120,RECFM=FB
ADAGLOB	The global parameter definitions. LRECL=80,RECFM=FB
ADAMES	Adabas Native SQL writes messages at preprocessing time to this dataset.
SYSUT1	A temporary work file. LRECL=80,BLKSIZE=3120,RECFM=FB
DDCARD	The Adabas parameters that are used by Adabas Native SQL at preprocessor time.
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 DD-names (in addition to any other user-defined datasets):

DDNAME	Description
DDCARD	The Adabas parameters that are used by the ADARUN module at run time.
DDPRINT	Adabas writes messages to this dataset at run time.
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. LRECL=80,BLKSIZE=3680,RECFM=FB