

Adabas Native SQL

Installation Manual for BS2000/OSD

Version 2.4.1

September 2009

This document applies to Adabas Native SQL Version 2.4.1 and to all subsequent releases.

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

Copyright © Software AG 2009. All rights reserved.

The name Software AG, webMethods and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA, Inc. Other company and product names mentioned herein may be trademarks of their respective owners.

Table of Contents

1	Installation Manual for BS2000/OSD	1
2	GENERAL INFORMATION	3
	Installation Jobs	4
	Using System Maintenance Aid	4
3	TAPE LAYOUT	5
4	INSTALLATION	9
	Copying the Tape Contents to Disk	10
	Relinking Adabas Native SQL	11
	Modifying the Procedure PSQL	12
	Modifying the Global Parameters	12
	Testing Adabas Native SQL	13
	File Names	15
5	INSTALLATION CHECKLIST	17
6	HINTS FOR USING ADABAS NATIVE SQL	19
	Periodic Groups and Multiple Fields	20
	Group Structure of Periodic Groups	20
	Dynamic Command-IDs	20
	Locating Errors	20
	Preprocessor Copy and Generate Facilities	20
	Restriction for MU Fields Within PE	21
	Hyphens and Break Characters in PL/I Programs	21
	READ ISN Statement	21
	PL/I - Margin Settings	21
	PL/I - Structure Variables in Superdescriptors	22
	Last Statement Restriction in COBOL/II	22
	Adabas Native SQL Preprocessor Condition Codes	22
	SQL0217 Error Message When Running the Precompiler	22
	Hints for Improving Adabas Native SQL Efficiency	23

1 Installation Manual for BS2000/OSD

This document describes how to install Adabas Native SQL on BS2000/OSD.

This information is structured into the following sections:

•	<i>General Information</i>	This section provides general information about the installation procedure.
•	<i>Tape Layout</i>	This section describes the contents of the distribution medium.
•	<i>Installation</i>	This section describes how to perform the installation.
•	<i>Installation Checklist</i>	This section describes how to verify that the installation has been successful.
•	<i>Hints for Using Adabas Native SQL</i>	This section contains a list of useful hints about using Adabas Native SQL.

2 GENERAL INFORMATION

- Installation Jobs 4
- Using System Maintenance Aid 4

This chapter covers the following topics:

Installation Jobs

The installation of Software AG products is performed by installation jobs. These jobs are either adapted "manually" or generated by System Maintenance Aid (SMA).

For each step of the installation procedure described below, the job number of a job performing the respective task is indicated. This job number refers to an installation job generated by SMA.

Using System Maintenance Aid

If you are using Software AG's System Maintenance Aid (SMA) for the installation process, refer to the System Maintenance Aid manual.

3 TAPE LAYOUT

Approximately 1900 pam pages of disk space will be required when installing Adabas Native SQL.

The volume serial number of the magnetic tape that includes Adabas Native SQL is shown clearly on the adhesive label affixed to the tape reel, and also in the *Report of Tape Creation* that accompanies the tape. Please check that these two numbers are identical.

Unless stated to the contrary, the tape contains the data sets listed below, which are discussed in the following sections.

Dataset Name on Tape	Dataset Name on Disk	Contents
SQL nnn .SRCE	SQL nnn .SRC	Source members and examples
SQL nnn .PAMS	SQL nnn .MOD	Module library
SQL nnn .ERRN	SQL nnn .ERRN	Error messages which may be required by Adabas Native SQL at preprocessing time

File SQL nnn .SRCE contains the following sources:

Name	Description
AEX1...AEX3	Ada examples using various Adabas Native SQL statements.
CEX1...CEX3	COBOL examples using various Adabas Native SQL statements.
FEX1...FEX3	FORTRAN examples using various Adabas Native SQL statements.
PEX1...PEX3	PL/I examples using various Adabas Native SQL statements.
PLINK	A sample procedure to link Adabas Native SQL. Use TSOSLNK to relink Adabas Native SQL.
PRTFLO	The source code of the routine that prints a flow-trace of all executed Adabas Native SQL statements during the execution of the program if MODE FLOW is set (FORTRAN).

Name	Description
PRTFLOP	The source code of the routine that prints a flow-trace of all executed Adabas Native SQL statements during the execution of the program if MODE FLOW is set (PL/I).
PRTFLOW	The source code of the routine that prints a flow-trace of all executed Adabas Native SQL statements during the execution of the program if MODE FLOW is set (COBOL).
PRTRAC	The source code of the routine that traces the record buffers and format buffers during the execution of the application program if MODE TRACE is set (FORTRAN).
PRTRACE	The source code of the routine that traces the record buffers and format buffers during the execution of the application program if MODE TRACE is set (COBOL).
PRTRACP	The source code of the routine that traces the record buffers and format buffers during the execution of the application program if MODE TRACE is set (PL/I).
PSQL	A sample procedure to preprocess, compile, link and execute an Ada, COBOL, FORTRAN or PL/I program.
RESPF	The source code of the response code interpretation routine (FORTRAN).
RESPINT	The source code of the response code interpretation routine (COBOL).
RESPPL1	The source code of the response code interpretation routine (PL/I).
SQFRDATE	Routine which converts format D number to numeric date.
SQFRTIME	Routine which converts format T number to numeric date and numeric time.
SQTODATE	Routine which converts numeric date to format D number.
SQTOTIME	Routine which converts numeric date and numeric time to format T number.
TYPESADA	Data definitions for use in Ada programs. This file must be compiled before using the Adabas Native SQL preprocessor with Ada source programs.

If necessary, the source modules may be edited to accommodate installation-specific requirements. The original source modules should be retained for future reference.

The module library (file SQL_{nnn}.PAMS on tape) contains the following members:

Name	Description
ADASQL	The Adabas Native SQL preprocessor (object).
APPEX1	An Adabas Native SQL module.
APPEX2	An Adabas Native SQL module.
APPEX3	An Adabas Native SQL module.
APPMSG	An Adabas Native SQL module.
APPTIME	An Adabas Native SQL module.
CMDIX	A module that generates the names of the record buffer.
FCID	Module that dynamically generates command-IDs during execution of the application program. Used in Ada and FORTRAN programs.
FINDCID	Module that dynamically generates command-IDs during execution of the application program. Used in COBOL and PL/I programs.

Name	Description
FJOBNAME	Module that gets the Job name.
PRPABEND	This module is used to abend an application program if an error occurred.
PRTFLOW	Module that prints a flow-trace of all executed Adabas Native SQL statements during the execution of the application program if MODE FLOW is set (object module, COBOL).
PRTRACE	Module that traces the record buffers and format buffers during the execution of the application program if MODE TRACE is set (object module, COBOL).
RESPINT	The response code interpretation routine (object module, COBOL).
RESPINT@	The response code interpretation routine constants.
SQFRDATE	Routine which converts format D number to numeric date.
SQFRTIME	Routine which converts format T number to numeric date and numeric time.
SQLPRE	The prelinked Adabas Native SQL module, but without an Adabas interface module.
SQTODATE	Routine which converts numeric date to format D number.
SQTOTIME	Routine which converts numeric date and numeric time to format T number.

4 INSTALLATION

- Copying the Tape Contents to Disk 10
- Relinking Adabas Native SQL 11
- Modifying the Procedure PSQL 12
- Modifying the Global Parameters 12
- Testing Adabas Native SQL 13
- File Names 15

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.

Version 2.2B of the COBOL compiler is a prerequisite for successful use of Adabas Native SQL. If you use an older version of the compiler, the object modules generated by your compiler may not be compatible with the Software AG-supplied routines RESPINT, PRTRACE and PRTFLOW; in this case, you must recompile these three routines from the source files supplied on the tape.

This chapter covers the following topics:

Copying the Tape Contents to 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, use the procedure described below. In this procedure, the values specified below must be supplied. To copy the datasets from tape to disk, perform the following steps:

Step 1: Copy the Library SRVnnn.LIB from Tape to Disk

This step is not necessary if you have already copied the library SRVnnn.LIB from another Software AG tape. For more information, refer to the element #READ-ME in this library. The library SRVnnn.LIB is stored on the tape as the sequential file SRVnnn.LIBS containing LMS commands. The current version nnn can be obtained from the Report of Tape Creation. To convert this sequential file into an LMS-library, execute the following commands:

```
/IMPORT-FILE SUPPORT=*TAPE(FILE-NAME=SRVnnn.LIBS, -  
/ VOLUME=<volser>, DEV-TYPE=<tape-device>)  
/ADD-FILE-LINK LINK-NAME=EDTSAM, FILE-NAME=SRVnnn.LIBS, -  
/ SUPPORT=*TAPE(FILE-SEQ=3), ACC-METH=*BY-CAT, -  
/ BUF-LEN=*BY-CAT, REC-FORM=*BY-CAT, REC-SIZE=*BY-CAT  
/START-EDT  
@READ '/'  
@SYSTEM 'REMOVE-FILE-LINK EDTSAM'  
@SYSTEM 'EXPORT-FILE FILE-NAME=SRVnnn.LIBS'  
@WRITE 'SRVnnn.LIBS'  
@HALT  
/ASS-SYSDTA SRVnnn.LIBS  
/MOD-JOB-SW ON=1  
/START-PROG $LMS  
/MOD-JOB-SW OFF=1  
/ASS-SYSDTA *PRIMARY
```

Where:

<tape-device>	is the device-type of the tape, e.g. TAPE-C4.
<volser>	is the VOLSER of the tape.

Step 2: Copy the Procedure COPY.PROC from Tape to Disk

To copy the procedure COPY.PROC to disk, call the procedure P.COPYTAPE in the library SRVnnn.LIB:

```
/CALL-PROCEDURE (SRVnnn.LIB,P.COPYTAPE), -
/ (VSNT=<volser>, DEVT=<tape-device>)
```

If you use a TAPE-C4 device, you may omit the parameter DEVT.

Step 3: Copy all Product Files from Tape to Disk

To copy all Software AG product files from tape to disk, enter the procedure COPY.PROC:

```
/ENTER-PROCEDURE COPY.PROC, DEVT=<tape-device>
```

If you use a TAPE-C4 device, you may omit the parameter DEVT. The result of this procedure is written to the file 'L.REPORT.SRV'.

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 PLINK can be used to link ADASQL.

The prelinked Adabas Native SQL module is in SQL_{nnn}.MOD, and it has the name SQLPRE.

The module ADAUSER reads ADARUN parameters via SYSDTA. If an alternative module is used, this dataset may not be needed.

Modifying the Procedure PSQL

Modify (edit) the supplied procedure PSQL as required and then execute the example programs. The following specifications in the procedures should be adapted to suit the system environment. The names of the procedure parameters are given in square brackets [], and the default values are given in parentheses ().

- The Adabas module library [&ADABAS] that contains the module ADARUN and the Adabas link modules (SAG.ADABAS.MOD);
- Language-specific files (e.g., TSOS.COB1.RTSMODLIB(&COBLIB) for COBOL; TSOS.FOR1.MODLIB(&FORLIB) for FORTRAN; TSOS.PLI1.MODLIB(&PL1LIB), TSOS.PLI1.TEXT.E(&PL1TEXT) for PL/I);
- The language [&LANG(COBOL)] in which the application program is written. This is specified by a procedure parameter coded as ADA or COBOL or FORT or PL1;
- The membername [&SOURCE] of the program to be preprocessed (CEX1);
- The following SQL precompile parameters: [&DB(1)] the DBID where the Predict data is kept; [&FDIC(11)] the file number of the Predict FDIC file; [&FNAT(8)] the file number of the FNAT file; [&ABORTFILE(8)] the file number for the ABORT parameter;
- The following ADARUN data: [&DB(1)] the DBID where the Predict data is kept; [&IDTNAME(ADABAS5B)] the IDTNAME where the DBID runs;
- [&SQLLIB(\$SAG.SQL.MOD)] the Adabas Native SQL module library;
- [&SQLPROC(\$SAG.SQL.CSQL)] the Adabas Native SQL executable module.

Modifying the Global Parameters

The global parameters for Adabas Native SQL, which are stored in the file D.ADAGLOB (FCB-TYPE=SAM, RECFORM=V), must also be modified.

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.

```

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 tape. The notation nnn represents the Version, Release and SM level of Predict at your site. Ensure that the relationships for soft coupling are defined in the data dictionary. 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 PSQL executes the following steps:

1. Preprocess the program with Adabas Native SQL;
2. Compile the preprocessed program (ADA, COBOL, FORTRAN or PL/I);
3. Link-edit the program. The following modules may be called by the user program at runtime:

COBOL	PL/I	Description
RESPINT (C)	RESPINT (C,P)	Error handling routine.
PRTRACE (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)	Generates dynamic Adabas command IDs. Activated if OPTION DYNAMCID is set.
PRPABEND (A)	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.

Ada	FORTRAN	Description
RESPF (F)	RESPF(F)	Error handling routine.
PRTRAC (F)	PRTRAC (F)	Runtime trace routine. Activated by the error handling routine if <code>MODE TRACE</code> is set.
PRTFLO (F)	PRTFLO (F)	Runtime Adabas Native SQL statement printing. Activated if <code>MODE FLOW</code> is set.
FCID (A)	FCID (A)	Generates dynamic Adabas command IDs. Activated if <code>OPTION DYNAMCID</code> 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:

- a. 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.
- b. Note: 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.
- c. 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.

Finally, the linked program (load module) is stored with the name *M.membername*;

4. Execute the program.

All listings generated by this procedure are given names starting with L. (SYSLST, SRCLINK, ERRLINK).

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.

File Names

The files used by Adabas Native SQL are listed below:

Name	Description
ADAGLOB	The global parameter definitions.
ADAIN	The source program to be preprocessed.
ADAMES	Adabas Native SQL writes messages at preprocessing time to this dataset.
ADAOUT	The output of the preprocessor. Embedded Adabas Native SQL statements in the source program appear in this file as comments and are followed by the generated Ada, COBOL, FORTRAN or PL/I code.
ADAOUT1	A temporary work file.
ADDOUT	A temporary work file.
DDLNKPAR	Defines the IDTNAME on which the Native SQL is running (for the ADALNK module.
SYSLST	Adabas Native SQL writes messages to this file if it abends.

These files are defined in the procedure PSQL. Each file name is prefixed with "D", for example D.ADAMES.

At run time, the execution of the Adabas Native SQL program requires the presence of a link name DDLIB which points to the Adabas Library containing the ADALNK. Also present, must be a link name DDLNKPAR pointing to a file containing: ADALNK IDTNAME=<idtname> where <idtname> is the BS2000 IDTNAME on which the database data to be accessed is available. The application program generated by Adabas Native SQL uses the following file names to refer to files (in addition to any other user-defined files):

Name	Description
SYSLST	If the FLOW facility is used, the flow-tracing output is written to this file.
TSTDMP	If the TRACE facility is used, the trace output is written to this file.

5

INSTALLATION CHECKLIST

- Verify that Predict Version 3.4 or later is installed.
- Verify that the files and fields that are to be used by Adabas Native SQL applications are documented in the data dictionary. Adabas Native SQL recognizes only the PREDICT file types *Adabas file* and *Adabas userview*. Other file types are ignored by the Adabas Native SQL preprocessor.
- Copy files from tape to disk either using SMA or job E.SQLTAPE in library SQL nnn .
- Verify that the procedure E.SAGI061 Step 2702 has been run to load the error messages.
- Edit the procedure PSQL as required.
- If an ADABAS interface module other than ADAUSER is to be used, relink ADABAS Native SQL using PLINK.
- Ensure that the standard test files EMPLOYEES and VEHICLES are available and that they are correctly documented in the data dictionary. In particular, ensure that the file numbers and the database ID are correct. Ensure that the relationships for soft coupling are defined in the data dictionary. If you wish to use FORTRAN, ensure that appropriate language synonyms are defined in the data dictionary (see Appendix B in the *Adabas Native SQL Reference Manual*).
- Run the procedure PSQL to execute the test programs. Verify the output.

6

HINTS FOR USING ADABAS NATIVE SQL

▪ Periodic Groups and Multiple Fields	20
▪ Group Structure of Periodic Groups	20
▪ Dynamic Command-IDs	20
▪ Locating Errors	20
▪ Preprocessor Copy and Generate Facilities	20
▪ Restriction for MU Fields Within PE	21
▪ Hyphens and Break Characters in PL/I Programs	21
▪ READ ISN Statement	21
▪ PL/I - Margin Settings	21
▪ PL/I - Structure Variables in Superdescriptors	22
▪ Last Statement Restriction in COBOL/II	22
▪ Adabas Native SQL Preprocessor Condition Codes	22
▪ SQL0217 Error Message When Running the Precompiler	22
▪ Hints for Improving Adabas Native SQL Efficiency	23

This chapter covers the following topics:

Periodic Groups and Multiple Fields

If the maximum number of occurrences of periodic groups or multiple fields is known, specify this number in Predict. Otherwise Adabas Native SQL will allocate buffers using the default maximum values, resulting in wasted storage.

Group Structure of Periodic Groups

Correct use of the `GROUP STRUCT` attribute in Predict can save significant amounts of space in the format buffers that are used when accessing records containing periodic groups. See *Defining More Attributes of Fields* in section *Field* of chapter *Predefined Object Types* in the *Predict Reference Manual*.

Dynamic Command-IDs

If the database is accessed from many modules within one linked program, the global parameter `OPTIONS DYNAMCID.` should be specified. This ensures that a unique command-ID will be generated for each Adabas command if the same cursor-name is used in more than one module. Be aware that this can degrade run-time performance. See *Improving Adabas Native SQL Efficiency* and the section on using command-IDs in the *Adabas Command Reference Manual*.

Locating Errors

The TRACE and FLOW facilities, which are switched on by means of global parameters, can be used to find runtime errors in the application program.

Preprocessor Copy and Generate Facilities

The Adabas Native SQL preprocessor supports the COPY and GENERATE statements. These are compatible with the implementation of the COPY and GENERATE statements in the Predict preprocessor, so it should not normally be necessary to use both Adabas Native SQL and Predict preprocessors.

Restriction for MU Fields Within PE

The data dictionary definition of a multiple-value field within a periodic group should not specify a counter field.

Hyphens and Break Characters in PL/I Programs

Field names as defined in the data dictionary may include hyphens ("-") and/or break characters ("_"). With Adabas Native SQL, each reference to a field name must match the definition in the data dictionary exactly. However, when generating PL/I output, Adabas Native SQL will change all hyphens found in data dictionary definitions to break characters, since hyphens are not valid in PL/I identifiers.

With previous versions of Adabas Native SQL, it was not necessary for references to field names to match the definitions in the data dictionary exactly: all break characters found in the source program were converted to hyphens before being compared with the data dictionary definitions; then, when generating PL/I output, Adabas Native SQL changed all hyphens to break characters.

The global parameter `UNDERSCORE NO.` can be used to ensure that Adabas Native SQL processes hyphens and break characters in a manner compatible with earlier versions.

READ ISN Statement

After issuing the `READ ISN` statement with `OPTIONS SEQUENCE`, the program should check for end-of-file (`ADACODE = 3`). In some applications, it may be necessary to compare the ISN of the record that was read with the ISN that was specified in the `WHERE` clause of the statement.

PL/I - Margin Settings

When using the PL/I compiler, the margins should be set to (2,72) (these are the default values).

PL/I - Structure Variables in Superdescriptors

In PL/I, it is not possible to assign a structure variable to a superdescriptor in the WHERE clause. A string variable should be used in place of the structure variable.

Last Statement Restriction in COBOL/II

The last statement in a COBOL/II program should not be an Adabas Native SQL statement. If necessary, the statement `EXIT.` can be coded at the end of the program.

Adabas Native SQL Preprocessor Condition Codes

The Adabas Native SQL preprocessor can set the following condition codes:

Code	Meaning
0	No error was detected
4	One or more warning messages issued during processing.
8	One or more errors found during processing.
12	Adabas Native SQL abended.

Programs generated by the Adabas Native SQL preprocessor can cause various errors at runtime. Adabas errors are detected by the generated code. In COBOL and PL/I programs, the Adabas response code as described in the *Adabas Messages and Codes Manual* is returned to the calling procedure. In FORTRAN programs, response code 8 is returned to the calling procedure.

SQL0217 Error Message When Running the Precompiler

This message is:

```
SQL0217 FDIC DBID OR FDIC FNR DOES NOT MATCH THE CONTROL RECORD
```

This message occurs if a DDM from another system is loaded into the Natural FDIC file and thereafter the DBID and FNR of the DDM are changed online to the current FDIC's DBID and FNR.

To rectify this, start the Natural containing the Predict:

```
LOGON SYSDIC
MENU
D      for defaults
A      for Adabas Native SQL
<Enter>
.      to exit
```

This will reset the Adabas Native SQL control record.

Hints for Improving Adabas Native SQL Efficiency

Reducing Database Accesses

Using the HISTOGRAM and READ statements as much as possible in preference to FIND can help to reduce the number of accesses made to the database.

Command IDs

An explicit Adabas command ID is assigned whenever a cursor is declared in an Adabas Native SQL statement. Adabas Native SQL statements that are performed repeatedly should have a cursor, since the command ID derived from the cursor name is used by Adabas to indicate that the format buffer need not be translated repeatedly.

Variable Indices

Periodic groups and multiple fields should only be referenced using variable indices when this is unavoidable, since variable indices cause additional format translations and also additional RC commands to be executed.

Record Buffers

The number of record buffers generated by Adabas Native SQL can be reduced by coding UPDATE and INSERT statements with the WHERE CURRENT OF *cursor-name* clause but without the SET clause. This only applies if the UPDATE or INSERT statement and the statement referenced by the cursor-name contain the same field structure.

SELECT Clause

Program efficiency is improved if only those fields that are needed are coded in the SELECT clause. You may be tempted to write "SELECT *" instead of coding the name of each field, but this can result in excessively large record buffers and slower-running programs.

Global Format ID

Application programs, in particular online application programs, should use the the global options parameter OPTIONS GFORMAT. This can improve efficiency by reducing the number of format buffer translations that are performed at runtime.