HINTS FOR USING ADABAS NATIVE SQL

This chapter covers the following topics:

- Periodic Groups and Multiple Fields

- Group Structure of Periodic Groups

- Dynamic Command-IDs

- Locating Errors

- Preprocessor Copy and Generate Facilities

- Restriction for MU Fields Within PE

- Hyphens and Break Characters in PL/I Programs

- READ ISN Statement

- PL/I - Margin Settings

- PL/I - Structure Variables in Superdescriptors

- Last Statement Restriction in COBOL/II

- Adabas Native SQL Preprocessor Condition Codes

- SQL0217 Error Message When Running the Precompiler

- Hints for Improving Adabas Native SQL Efficiency

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.