

Preparing Programs for Static Execution

This section describes how to prepare Natural programs for static execution:

- Basic Principles
- Generation Procedure: CMD CREATE Command
- Precompilation of the Generated Assembler Program
- Modification Procedure: CMD MODIFY Command
- BIND of the Precompiled DBRM

Note:

For an explanation of the symbols used in this section to describe the syntax of Natural statements, see Syntax Symbols in the Natural Statements documentation.

Basic Principles

Static SQL is generated in Natural batch mode for one or more Natural applications which can consist of one or more Natural object programs. The number of programs that can be modified for static execution in one run of the generation procedure is limited to 999.

During the generation procedure, the database access statements contained in the specified Natural objects are extracted, written to work files, and transformed into a temporary Assembler program. If no Natural program is found that contains SQL access or if any error occurs during static SQL generation, batch Natural terminates and condition code 40 is returned, which means that all further JCL steps must no longer be executed.

The Natural modules NDBCHNK and NDBSTAT must reside in a steplib of the generation step. Both are loaded dynamically during the execution of the generation step.

The temporary Assembler program is written to a temporary file (the Natural work file CMWKF06) and precompiled. The size of the workfile is proportional to the maximum number of programs, the number of SQL statements and the number of variables used in the SQL statements. During the precompilation step, a database request module (DBRM) is created, and after the precompilation step, the precompiler output is extracted from the Assembler program and written to the corresponding Natural objects, which means that the Natural objects are modified (prepared) for static execution. The temporary Assembler program is no longer used and deleted.

A static DBRM is created by using either the sample job provided on the installation tape or an appropriate job created with the Create DBRM function.

Generation Procedure: CMD CREATE Command

To generate static SQL for Natural programs, LOGON to the Natural system library SYSDB2.

Since a new SYSDDB2 library has been created when installing Natural for DB2, ensure that it contains all Predict interface programs necessary to run the static SQL generation. These programs are loaded into SYSDDB2 at Predict installation time (see the relevant Predict documentation).

Then specify the CMD CREATE command and the Natural input necessary for the static SQL generation process; the CMD CREATE command has the following syntax:

```
CMD CREATE DBRM static-name USING using-clause
{ application-name,object-name,excluded-object }
:
:
```

The generation procedure reads but does not modify the specified Natural objects. If one of the specified programs was not found or had no SQL access, return code 4 is returned at the end of the generation step.

Static Name

If the PREDICT DOCUMENTATION option is to be used, a corresponding Predict static SQL entry must be available and the *static-name* must correspond to the name of this entry. In addition, the *static-name* must correspond to the name of the DBRM to be created during precompilation. The *static-name* can be up to 8 characters long and must conform to Assembler naming conventions.

USING-Clause

The *using-clause* specifies the Natural objects to be contained in the DBRM. These objects can either be specified explicitly as INPUT DATA in the JCL or obtained as PREDICT DOCUMENTATION from Predict.

```
{ INPUT DATA
  PREDICT DOCUMENTATION } [ WITH XREF { YES
  { NO
  { FORCE } } ] [ FS { OFF } ] [ LIB lib-name ]
```

If the parameters to be specified do not fit in one line, specify the command identifier (CMD) and the various parameters in separate lines and use both the input delimiter (as specified with the ID parameter; default is a comma (,)) and the continuation indicator (as specified with the CF parameter; default is a percent (%) character as shown in the following example:

Example:

```
CMD
CREATE,DBRM,static,USING,PREDICT,DOCUMENTATION,WITH,XREF,NO,%
LIB,library
```

Alternatively, you can also use abbreviations as shown in the following example:

Example:

```
CMD CRE DBRM static US IN DA W XR Y FS OFF LIB library
```

The sequence of the parameters USING, WITH, FS, and LIB is optional.

INPUT DATA

As input data, the applications and names of the Natural objects to be included in the DBRM must be specified in the subsequent lines of the job stream (*application-name,object-name*). A subset of these objects can also be excluded again (*excluded-objects*). Objects in libraries whose names begin with SYS can be used for static generation, too.

The applications and names of Natural objects must be separated by the input delimiter (as specified with the ID parameter; default is a comma (,)). If you wish to specify all objects whose names begin with a specific string of characters, use an *object-name* or *excluded-objects* name that ends with asterisk notation (*). To specify all objects in an application, use asterisk notation only.

Example:

```
LIB1,ABC*
  LIB2,A*,AB*
  LIB2,*
  :
  .
```

The specification of applications/objects must be terminated by a line that contains a period (.) only.

PREDICT DOCUMENTATION

Since Predict supports static SQL for DB2, you can also have Predict supply the input data for creating static SQL by using already existing PREDICT DOCUMENTATION.

WITH XREF Option

Since Predict Active References supports static SQL for DB2, the generated static DBRM can be documented in Predict and the documentation can be used and updated with Natural.

WITH XREF is the option which enables you to store cross-reference data for a static SQL entry in Predict each time a static DBRM is created (YES). You can instead specify that no cross-reference data are stored (NO) or that a check is made to determine whether a Predict static SQL entry for this static DBRM already exists (FORCE). If so, cross-reference data are stored; if not, the creation of the static DBRM is not allowed. For more detailed information on Predict Active References, refer to the relevant Predict documentation.

When WITH XREF (YES/FORCE) is specified, XREF data are written for both the Predict static SQL entry (if defined in Predict) and each generated static Natural program. However, static generation with WITH XREF (YES/FORCE) is possible only if the corresponding Natural programs have been cataloged with XREF ON.

WITH XREF FORCE only applies to the USING INPUT DATA option.

Note:

If you do not use Predict, the XREF option must be omitted or set to NO and the module NATXRF2 need not be linked to the Natural nucleus.

FS Option

If the FS (file server) option is set to ON, a second SELECT is generated for the Natural file server. ON is the default setting.

If the FS option is set to OFF, no second SELECT is generated, which results in less SQL statements being generated in your static DBRM and thus in a smaller DBRM.

LIB Option

With the LIB (library) option, a Predict library other than the default library (*SYSSTA*) can be specified to contain the Predict static SQL entry and XREF data. The name of the library can be up to eight characters long.

Precompilation of the Generated Assembler Program

In this step, the precompiler is invoked to precompile the generated temporary Assembler program. The precompiler output consists of the DBRM and a precompiled temporary Assembler program which contains all the database access statements transformed from SQL into Assembler statements.

Later, the DBRM serves as input for the BIND step and the Assembler program as input for the modification step.

Modification Procedure: CMD MODIFY Command

The modification procedure modifies the Natural objects involved by writing precompiler information into the object and by marking the object header with the *static-name* as specified with the CMD CREATE command.

In addition, any existing copies of these objects in the Natural global buffer pool (if available) are deleted and XREF data are written to Predict (if specified during the generation procedure).

To perform the modification procedure, LOGON to the Natural system library SYSDB2 and specify the CMD MODIFY command which has the following syntax:

```
CMD MODIFY [ XREF ]
```

The input for the modify step is the precompiler output which must reside on a dataset defined as the Natural work file CMWKF01.

The output consists of precompiler information which is written to the corresponding Natural objects. In addition, a message is returned telling you whether it was the first time an object was modified for static execution (modified) or whether it had been modified before (re-modified).

Assembler/Natural Cross-References

If you specify the XREF option of the MODIFY command, an output listing is created on the work file CMWKF02, which contains the DBRM name and the Assembler statement number of each statically generated SQL statement together with the corresponding Natural source code line number, program name, library name, database ID and file number.

DBRMNAME	STMTNO	LINE	NATPROG	NATLIB	DB	FNR	COMMENT
TESTDBRM	000627	0390	TESTPROG	SAG	010	042	INSERT
	000641	0430					INSERT
	000652	0510					SELECT
	000674	0570					SELECT
	000698	0570					SELECT 2ND
	000728	0650					UPD/DEL
	000738	0650					UPD/DEL 2ND
	000751	0700					SELECT
	000775	0700					SELECT 2ND

Column	Explanation
DBRMNAME	Name of the DBRM which contains the static SQL statement.
STMTNO	Assembler statement number of the static SQL statement.
LINE	Corresponding Natural source code line number.
NATPROG	Name of the Natural program that contains the static SQL statement.
NATLIB	Name of the Natural library that contains the Natural program.
DB / FNR	Natural database ID and file number.
COMMENT	Type of SQL statement, where 2ND indicates that the corresponding statement is used for a reselection; see also the Concept of the File Server.

BIND of the Precompiled DBRM

It is recommended to run the BIND job *after* the MODIFY job.

This step performs the BIND against DB2. One or more DBRMs (as created by the precompiler) are processed to create a DB2 application plan. In addition to the static DBRMs created above, this application plan contains the dynamic DBRM NDBIOMO provided by Natural itself.

A DBRM can be bound into any number of application plans where it might be required. A plan is physically independent of the environment where the program is to be run. However, you can group your DBRMs logically into plans which are to be used for either batch or online processing, where the same DBRM can be part of both a batch plan and an online plan.

Unless you are using plan switching, only one plan can be executed per Natural session. Thus, you must ensure that the plan name specified in the BIND step is the same as the one used to execute Natural.