Installing Natural for SQL/DS

This section describes step by step how to install the Natural interface to SQL/DS (in the remainder of this section also referred to as NSQ).

This section covers the following topics:

- Installation Jobs
- Using System Maintenance Aid
- Prerequisites
- Installation under CMS
- Installation under VSE/ESA
- Installation Verification
- Natural Parameter Modification for SQL/DS
- Parameter Module NDBPARM

Installation Jobs

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

For each step of the installation procedures described later in this section, the job number of a job performing the respective task is indicated. This job number refers to an installation job generated by SMA. If you are not using SMA, an example job of the same number is provided in the job library on the NSQ installation tape; you must adapt this example job to your requirements. That the job numbers on the tape are preceded by a product code (for example, NSQI070).

Using System Maintenance Aid

For information on using Software AG’s System Maintenance Aid for the installation process, refer to the System Maintenance Aid documentation.

Prerequisites

- Base Natural Version 3.1 or above must be installed first; you cannot install Natural 3.1 and Natural for SQL/DS 3.1 at the same time.
- The Software AG Editor must be installed (as described in the Natural Installation Guide for Mainframes).
- If you want to share your FUSER system file between Natural 2.2 and Natural 3.1, INPL update tape NQ3405 must be applied to your Natural 2.2 environment.
- SQL/DS Version 3.5 or higher is required.

Further product/version dependencies are specified under Natural and Other Software AG Products and Operating/Teleprocessing Systems Required in the current Natural Release Notes for Mainframes.

Installation under CMS

This section only applies to the installation of NSQ under CMS.

Copyright Software AG 2002

Installation Tape

The installation tape was created under OS/390; it has standard OS/390 labels and headers. It contains the datasets listed in the table below. The sequence of the datasets is shown in the Report of Tape Creation which accompanies the installation tape.

Dataset Name	Contents
NSQ*nnn*.TAPE	NSQ source modules, load modules and installation EXECs. This dataset is in TAPE DUMP format and must be loaded onto the installation minidisk.
NSQ*nnn*.INPL	NSQ utility programs in INPL format.
NSQ*nnn*.ERRN	NSQ error messages.

The notation *nnn* in dataset names represents the version number of the product.

Copying the Tape Contents to Disk

The tape file NSQ*nnn*.TAPE was created with the CMS TAPE DUMP facility. Load the contents of the tape to your A-disk. The free space should be at least 450 4-KB blocks; for example, 3 cylinders on 3350 or 3380 disks.

Ask the system operator to attach a tape drive to your virtual machine at address X’181’ and mount the NSQ installation tape.

To position the tape for the TAPE LOAD command, calculate the number of tape marks as follows: If the sequence number of NSQ*nnn*.TAPE - as shown by the Report of Tape Creation - is *n*, you must position over 3*n*-2 tape marks; that is, FSF 1 for the first dataset, FSF 4 for the second, etc.

Position the tape by issuing the CMS command:

TAPE FSF *fsfs*

where *fsfs* is calculated as described above.

Load the NSQ installation material by issuing the CMS command:

TAPE LOAD * * A

You may wish to keep the tape drive attached to your virtual machine, because the tape is still needed in Step 7 of the installation procedure.

Preparing the Installation

Perform the following steps to prepare the installation of NSQ:

1. Ensure that the required SQL/DS database machine is activated in multiple- user mode and that the user machine for this installation is properly configured and initialized to access the SQL/DS database machine.
2. All precompilations as well as NSQ itself take advantage of the implicit CONNECT mechanism provided by VM. Therefore, ensure that the VM user ID is authorized for SQL/DS.
3. Ensure that your user machine has access to the following minidisks:
   the SQL/DS production minidisk,
   the Natural Version 3.1 installation minidisk.
4. Ensure that the Adabas environment for your user machine is set up.

Concerning the following installation steps, also refer to the section Installing Natural under VM/CMS in the Natural Installation Guide for Mainframes.

 Copyright Software AG 2002

Installation Procedure

Perform the following steps to install NSQ:

Step 1: Generate the NSQ I/O module NDBIOMO

Generate NDBIOMO by using the command:

GENIOMO SQL/DS *n*

GENIOMO generates the assembly source for NDBIOMO from the existing source NDBIOTM. It prompts you for the Natural/CMS batch module and invokes the Natural program NDBGENI, which is loaded with INPL during the base Natural Version 3.1 installation.

GENIOMO is invoked with the following two parameters:

- the DB-environment parameter, which must be set to SQL/DS,
- the parameter "*n*" to specify the number of statements for dynamic access; the default value is 10.

NDBIOMO performs the dynamic access to SQL/DS and contains all necessary EXEC SQL statements. In addition, it contains some special SQL statements which cannot be executed in dynamic mode.

An output report is created by this job and should be checked for successful completion. In addition, a condition code of 0 indicates normal completion.

Step 2: Precompile and assemble NDBIOMO

Precompile and assemble NDBIOMO using the command:

NDBIOMO

Note:
Since no precompiler options are specified, the default SQL/DS isolation level "Repeatable Read" may lead to locking problems, because all SQL/DS locks are held until the end of the transaction. Thus, depending on your application, it may be necessary to specify a different isolation level.

Step 3: Modify and assemble the NSQ parameter module NDBPARM

Assemble NDBPARM using the command:

NDBPARM

The NSQ parameter module contains the macro NDBPRM, which contains parameters specific to the Natural interface to SQL/DS.

You can generally use the default values for all parameters. Modify only the values of the parameters whose default values do not suit your requirements.

The individual parameters are described in the section The Parameter Module NDBPARM.

Step 4: Modify NATPARM

Adapt your Natural parameter module NATPARM by adding parameters specific to Natural for SQL/DS as described in the section Natural Parameter Modification for SQL/DS.

Copyright Software AG 2002

Step 5: Modify NAT$LOAD LOADLIST

Edit the member NAT$LOAD EXEC provided on the Natural/CMS installation tape and add the following line to the existing LOADLIST statements:

```
LOADLIST = LOADLIST 'NDBNUC NDBNSQ NDBPARM NDBIOMO ARIRVSTC'
```

Step 6: Generate a Natural module

Generate the Natural/CMS load module using the command:

NATBLDM

NATBLDM is provided on the Natural CMS/installation tape and prompts you for the name of the Natural nucleus and generates the executable Natural module.

 Copyright Software AG 2002

Step 7: Load Natural objects and error messages into system file

In this step, the NSQ system programs, maps and DDMs (dataset NSQ*nnn*.INPL) and the NSQ error messages file (dataset NSQ*nnn*.ERRN) are loaded into the Natural system file.

If the tape drive used when copying the contents of the installation tape to disk was detached from your virtual machine, ask the system operator to attach a tape drive to your virtual machine at address X'181' and mount the Natural installation tape.

Issue the following command:

NSQINPL

You are prompted for the name of the command to invoke Natural. Enter the name of the Natural module generated in Step 6.

NSQINPL then positions the tape and loads the Natural objects and error messages.

The INPL job loads objects into the libraries SYSDDM, SYSTEM and SYSSQL.

The ERRLODUS job loads error messages into the library SYSERR.

The NSQ system programs and error messages **must** be loaded into the Natural 3.1. FNAT system file

Warning:
Ensure that your newly created SYSSQL library contains all necessary Predict interface programs, which are loaded into SYSSQL when installing Predict (see the relevant Predict documentation).

Installation under VSE/ESA

Under VSE/ESA, Natural for SQL/DS basically consists of two parts:

1. An environment-independent nucleus, which can be linked to a shared Natural nucleus and loaded in the shared virtual area (SVA) of the operating system.
2. Environment-dependent components, which must be linked to the appropriate Natural environment-dependent interface.

This section covers the following topics:

- Installation Tape
- Installation Procedure

Installation Tape

The installation tape contains the datasets listed in the table below. The sequence of the datasets is shown in the Report of Tape Creation which accompanies the installation tape.

Dataset Name	Contents
NSQ*nnn*.LIBR	LIBR backup file.
NSQ*nnn*.INPL	NSQ utility programs in INPL format.
NSQ*nnn*.ERRN	NSQ error messages.

The notation *nnn* in dataset names represents the version number of the product.

Copying the Tape Contents to Disk

Copy the sublibrary containing the sample installation jobs from tape using the following JCS:

```
* $$ JOB JNM=NATJOBS,CLASS=0,DISP=D,LDEST=*,SYSID=1
* $$ LST CLASS=A,DISP=D
// JOB NATJOBS
// ASSGN SYS005,IGN
// ASSGN SYS006,cuu,VOL=Tnnnnn
// MTC REW,cuu
// MTC FSF,SYS006,nn
* Tape positioned at tape mark nn
*  *** NOW PROCESSING NSQnnn.LIBR - SUBLIBRARY NSQnnnJ ***
// EXEC LIBR,PARM='MSHP'
RESTORE SUBLIB=SAGLIB.NSQnnnJ:SAGLIB.NSQnnnJ -
TAPE=SYS006 -
LIST=YES -
REPLACE=NO
/*
// MTC REW,SYS006
/*
/&
* $$ EOJ
```

 Copyright Software AG 2002

Notation:

cuu	represents the physical unit address of the tape drive.
nn	represents the file sequence number as shown in the Report of Tape Creation.
nnn	represents the version number of the product.

If you are not using System Maintenance Aid, adapt and run job NSQTAPE to copy the dataset from tape to disk. NSQTAPE is contained in sublibrary NSQ*nnn*J on the Natural installation tape.

The dataset type and the space it requires on disk are shown in the Report of Tape Creation.

Copyright Software AG 2002

Installation Procedure

The following steps describe the procedure for installing the components of NSQ.

Step 1: Generate the NSQ I/O Module NDBIOMO - Job I055, Step 1600

By executing a standard Natural batch job, this step generates the assembly source for NDBIOMO from the member NDBIOTM.

This batch job invokes the Natural program NDBGENI, which is loaded INPL during the base Natural installation. NDBGENI contains the following two parameters, which can be modified to meet your specific requirements:

- the DB-environment parameter, which must be set to SQL/DS,
- the parameter to specify the number of statements for dynamic access.

NDBIOMO performs the dynamic access to SQL/DS and contains all necessary EXEC SQL statements (see further information on NDBIOMO in the section Internal Handling of Dynamic Statements). In addition, it contains some special SQL statements which cannot be executed in dynamic mode.

An output report is created by this job and should be checked for successful completion. In addition, a condition code of 0 indicates normal completion.

Step 2: Precompile and Assemble NDBIOMO - Job I055, Steps 1610 and 1620

Precompile (using the SQL precompiler) and assemble NDBIOMO. Ensure that an appropriate SQL/DS user ID and password is specified for precompiling.

Note:
Since no precompiler options are specified, the default SQL/DS isolation level "Repeatable Read" may lead to locking problems, because all SQL/DS locks are held until the end of the transaction. Thus, depending on your application, it may be necessary to specify a different isolation level.

Step 3: Modify and Assemble the NSQ Parameter Module NDBPARM - Job I055, Step 1640

The NSQ parameter module contains the macro NDBPRM with parameters specific to the Natural interface to SQL/DS.

You can generally use the default values for all parameters. Modify only the values of the parameters whose default values do not suit your requirements.

The individual parameters are described in the section The Parameter Module NDBPARM.

Step 4: Modify and Reassemble NATPARM

Adapt your Natural parameter module NATPARM by adding parameters specific to Natural for SQL/DS and reassemble NATPARM.

Step 5: Relink your Natural Nucleus

Modify the JCL used to link your Natural nucleus by adding the following INCLUDE cards and the corresponding DLBL statements:

 Copyright Software AG 2002

INCLUDE NDBNUC	Environment-independent NSQ nucleus
INCLUDE NDBNSQ	Environment-independent SQL/DS interface
INCLUDE NDBPARM	NSQ parameter module created in Step 3
INCLUDE NDBIOMO	NSQ I/O module created in Step 1
INCLUDE *xxxxxxx*	Environment-dependent SQL/DS interface (see below)

Depending on your environment(s), INCLUDE the appropriate environment-specific language interface "*xxxxxxxx*" as shown in the following table:

Interface	**Environment**
ARIPRDID	In batch mode
ARIRRTED	Under CICS

Note:
If you want to use NSQ in both environments, repeat this step for each of these environments.

Instead of link-editing your Natural nucleus in the way described above, you have the following alternatives:

1. If you use a shared Natural nucleus, only include NDBNUC and NDBNSQ in the link-edit of this nucleus. All other modules must be included in the link-edit of your Natural environment-dependent nucleus.
2. Remove NDBNUC and NDBNSQ from the link-edit of the Natural nucleus and link-edit them as a separate module with the mandatory *entry* name NATGWDB2. The *name* of the resulting phase is arbitrary. However, if you use a name different from NATGWDB2, this name must be specified as an alias name in an NTALIAS macro entry of the Natural parameter module. This way of link-editing only applies if the Natural Resolve CSTATIC Addresses feature (RCA) is used.
3. Include all modules in the link-edit job of a separate Natural parameter module with the mandatory *entry* name CMPRMTB. The *name* of the resulting phase is arbitrary. This way of link-editing only applies if an alternative parameter module (PARM profile parameter) is used.
4. If link-editing is done in this way, you can install NDB without having to modify your Natural nucleus or driver.

If link-editing is done according to number 2. or 3., the following applies:

Under CICS:
the resulting module must be defined via a PPT entry or RDO:

```
DFHPPT TYPE=ENTRY,PROGRAM=module-name,PGMLANG=ASSEMBLER
```

Step 6: Load Natural Objects Into System File - Job I061, Step 1600

In this step, the NSQ system programs, maps and DDMs are loaded into the Natural system files. The INPL job loads objects into the libraries SYSDDM, SYSTEM and SYSSQL.

The NSQ system programs **must** be loaded into the Natural 3.1 FNAT system file.

Warning:
Ensure that your newly created SYSSQL library contains all necessary Predict interface programs, which are loaded into SYSSQL when installing Predict (see the relevant Predict documentation).

Step 7: Load Natural Error Messages into System File - Job I061, Step 1620

This step executes a batch Natural job that runs an error load program using the NSQ*nnn*.ERRN dataset as input. The ERRLODUS job loads error messages into the library SYSERR on the FNAT system file.

The NSQ error messages **must** be loaded into the Natural 3.1 FNAT system file.

Installation Verification

This section covers the following topics:

- Prepare your SQL/DS Environment
- Online Verification Methods
- Sample Batch Verification Job - VSE/ESA only

Prepare your SQL/DS Environment

As all dynamic access to SQL/DS is performed by NDBIOMO, all NSQ users must have RUN privilege on NDBIOMO.

Online Verification Methods

To verify the installation of the Natural interface to SQL/DS online, you can use either of the following methods:

- SQL Services
- DEM* Sample Programs

 Copyright Software AG 2002

SQL Services

Perform the following steps to verify and check the installation of NSQ using the SQL Services of the Natural utility SYSDDM.

1. Invoke Natural.
2. Invoke SYSDDM.
   On the SYSDDM main menu enter function code "B" to invoke the SQL Services function.
   Enter function code "S" and specify SQL system "SQL/DS" to select all SQL/DS tables.
   The communication between Natural and SQL/DS works if all existing SQL/DS tables are displayed.
3. For one of the tables, generate a Natural DDM as described in the section Generate DDM from an SQL Table.
   To enable SYSDDM to generate a DDM, the Natural administrator requires access to the following SQL/DS tables:

SYSTEM.SYSCATALOG
SYSTEM.SYSCOLUMNS
SYSTEM.SYSINDEXES
SYSTEM.SYSVIEWS
SYSTEM.SYSSYNONYMS
SYSTEM.SYSUSAGE

4. After you have generated a DDM, access the corresponding SQL/DS table with a simple Natural program:

Example:

```
FIND view-name WITH field = value
   DISPLAY field
  LOOP
  END
```

5. If you receive the message SYSFUL 3700, enter the command SQLERR to display the corresponding SQL return code. See the description of the SQLERR command.

DEM2* Sample Programs

To verify and test your installation you can also use the sample programs DEM2* in the library SYSSQL provided on the installation tape.

Using these sample programs, you can create an SQL/DS table using DEM2CREA and create the corresponding DDM via SYSDDM. You can then store data in the created table using DEM2STOR and retrieve data from the table using DEM2FIND or DEM2SEL. You can also drop the table using program DEM2DROP.

Copyright Software AG 2002

Sample Batch Verification Job - VSE/ESA only

To verify the installation of the Natural interface to SQL/DS, a sample batch verification job (Job I065) is provided. This step contains sample JCL and sample programs to test Natural with NSQ in batch mode.

The sample program DEM2CONN performs the connection to the database, which is required before you can run a Natural program that accesses SQL/DS. DEM2CONN calls the DB2SERV module with function code "U" which in turn calls the database connect services.

Sample program DEM2JOIN performs a JOIN combining information from SQL/DS SYSTEM.SYSDBSPACE and SYSTEM.SYSCATALOG.

 Copyright Software AG 2002

Natural Parameter Modification for SQL/DS

This section covers the following topics:

- DB2SIZE Parameter
- NTDB Macro
- Performance Considerations for the DB2SIZE Parameter

DB2SIZE Parameter

Add the following Natural profile parameter to your NATPARM module:

DB2SIZE=*nn*

The DB2SIZE parameter can also be specified dynamically. It indicates the size of the SQL/DS buffer area, which should be set to at least 6 KB.

The setting of DB2SIZE can be calculated according to the following formula:

((808 + *n1* *** 40 +** *n*2 *** 72) + 1023) / 1024 KB**

The variables *n1* and *n2* correspond to:

n1	the number of statements for dynamic access as specified as the second parameter in job I055, step 1600 (under VSE/ESA).
n2	the maximum number of nested database loops as specified with the MAXLOOP parameter in NDBPARM.

Note:
Ensure that you have also added the Natural parameters required for the Software AG Editor (see Installing the Software AG Editor in the Natural Installation Guide for Mainframes).

Since DB2SIZE applies to Natural for SQL/DS and Natural for DB2, it should be set to the maximum value if you run more than one of these environments.

Copyright Software AG 2002

NTDB Macro

Add an NTDB macro for database type SQL specifying the list of logical database numbers that relate to SQL/DS tables. All Natural DDMs that refer to an SQL/DS table must be cataloged with a DBID from this list.

DBIDs can be any number from 1 to 254; a maximum of 254 entries can be specified. For most user environments, one entry is sufficient.

Note:
Ensure that all NSQ DDMs used when cataloging a given program have a valid SQL/DS DBID. Also ensure that the DBIDs selected in the NTDB macro for SQL/DS do not conflict with DBIDs selected for other database systems.

As of base Natural Version 2.2, the DBID for SQL/DS used when cataloging a Natural program does no longer have to be in the NTDB list of DBIDs used when executing this program. Therefore, when executing existing Natural programs, DBID 250 is no longer mandatory. It is, however, still strongly recommended when cataloging Natural programs, because so far all DDMs have been cataloged with DBID 250.

Two sample NTDB macros follow:

```
NTDB SQL,250
NTDB SQL,(200,250,251)
```

 Copyright Software AG 2002

Performance Considerations for the DB2SIZE Parameter

During execution of an SQL statement, storage is allocated dynamically to build the SQLDA for passing the host variables to SQL/DS.

In previous Natural for SQL/DS versions, this storage was always obtained from the TP monitor or operating system. For performance reasons, it is now first attempted to meet the storage requirements by free space in the Natural for SQL/DS buffer (DB2SIZE). Only if there is not enough space available in this buffer, the TP monitor or operating system is invoked.

To take advantage of this performance enhancement, you must specify your DB2SIZE larger than calculated according to the formula. The additional storage requirements (in bytes) can be calculated as follows:

- With sending fields:
  64 + *n* * 56
  where "*n*" is the number of sending fields in an SQL statement.
  The storage is freed immediately after the execution of the SQL statement.
- With receiving fields (that is, with variables of the INTO list of a SELECT statement):
  64 + *n* * 56 + 24 + *n* * 2
  where "*n*" is the number of receiving fields in an SQL statement.
  The storage remains allocated until the loop is terminated.

Example:

If you use the default value 10 for both variables (*n1* and *n2*), the calculated DB2SIZE will be 5896 bytes. However, if you specify a DB2SIZE of 20 KB, the available space for dynamically allocated storage will be 14584 bytes, which means enough space for up to either 259 sending fields or 249 receiving fields.

As space for receiving fields remains allocated until a database loop is terminated, the number of fields that can be used inside such a loop is reduced accordingly: for example, if you retrieve 200 fields, you can update about 50 fields inside the loop.

Note:
When using VARCHAR fields (that is, fields with either an accompanying L@ field in the Natural view or an explicit LINDICATOR clause), additional storage is allocated dynamically if the L@ or LINDICATOR field is not specified directly in front of the corresponding base field. Therefore, always specify these fields in front of their base fields.

Copyright Software AG 2002

Parameter Module NDBPARM

The NDBPARM source module contains Natural parameters specific to an SQL/DS environment. The parameter default values can be modified to meet site-specific requirements (see Step 3 of the installation procedures).

The individual parameters are described below. Their values cannot be dynamically overwritten.

NDBPARM contains the following parameters:

Parameter	Function
BTIGN	Ignores errors which result from BACKOUT TRANSACTION statements that are issued too late.
CONVERS	Allows conversational mode under CICS.
MAXLOOP	Specifies the maximum number of nested program loops.
STATDYN	Allows dynamic execution of statically generated SQL statements if the static execution returns an error.

BTIGN - Ignore Error after late BACKOUT TRANSACTION

This parameter is relevant in CICS environments only.

This parameter is used to ignore the error which occurs after a BACKOUT TRANSACTION statement that came too late to backout the current transaction, because of an implicit syncpoint issued by the TP monitor.

Possible Value	Default Value	Explanation
ON	ON	The error after a late BACKOUT TRANSACTION is ignored.
OFF		The error after a late BACKOUT TRANSACTION is **not** ignored.

CONVERS - Allows Conversational Mode under CICS

This parameter is used to allow conversational mode in CICS environments.

Possible value	Default value	Explanation
ON	ON	Conversational mode is allowed.
OFF		Conversational mode is **not** allowed.

If this parameter is set to OFF, you cannot continue database loops across terminal I/Os; if so, the SQL codes -501, 504, 507, 514 or 518 may occur.

If you use the SYSDDM SQL services in a CICS environment, specify CONVERS=ON, otherwise the above mentioned errors could occur. See also the section SQL Services.

 Copyright Software AG 2002

MAXLOOP - Maximum Number of Nested Database Loops

This parameter specifies the maximum possible number of nested database loops.

Possible values	Default value
1 - 99	10

STATDYN - Allow Static to Dynamic Switch

This parameter is used to allow dynamic execution of statically generated SQL statements if static execution returns an error.

PossibleValue	Default value	Explanation
NEVER	NEVER	Dynamic execution is never allowed.
ALWAYS		Dynamic execution is always allowed after an error.
SPECIAL		Dynamic execution is allowed after special errors only. These special errors are: NAT3706 Load module not found. SQL -805 SQL/DS package does not exist. SQL -818 Mismatch of timestamps.