

Installing Natural for DB2

This section describes how to install the Natural interface to DB2 (in the remainder of this section also referred to as NDB) in the various environments supported.

The installation procedures contain a number of options that depend on the TP monitor being used as well as on other site requirements.

- Installation Jobs
 - Using System Maintenance Aid
 - Prerequisites
 - Installation Tape
 - Installation Procedure
 - Installation Verification
 - Natural Parameter Modification for DB2
 - Parameter Module NDBPARAM
 - Special Requirements for Natural Tools for DB2
 - Natural for DB2 Server Stub
-

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 procedure 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 NDB installation tape; you must adapt this example job to your requirements. Note that the job numbers on the tape are preceded by a product code (for example, NDBI070).

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 DB2 Version 3.1 at the same time.
- The Software AG Editor must be installed (see Installing the Software AG Editor in the Natural Installation Guide for Mainframes).
- If you want to use the DB2 DL/I batch support (DSNMTV01), the Natural interface to DL/I is required.
- If you want to share your FUSER system file between Natural 2.2 and Natural 3.1. INPL update tape NQ3404 must be applied to your Natural 2.2 environment.

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 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
NDB nnn .SRCE	NDB source modules
NDB nnn .LOAD	NDB load modules
NDB nnn .INPL	NDB utility programs in INPL format
NDB nnn .ERRN	NDB error messages
NDB nnn .JOBS	NDB installation jobs

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

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, follow the instructions below.

This section explains how to:

- Copy data set COPY.JOB from tape 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 tape to disk.

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

The data set COPY.JOB (label 2) contains the JCL to unload all other existing data sets from tape 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 tape 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 tape to your disk.

Installation Procedure

This section describes how to install Natural for DB2 in various environments:

1. Steps Common to all Environments
2. Steps Specific to
 - a CICS Environment
 - a Com-plete Environment
 - an IMS/TM Environment
 - a TSO Environment

Steps Common to all Environments

The following steps describe the procedure for installing the components of NDB that are common to all environments.

Step 1: Allocate a DBRM library for use with NDB

Allocate a PDS as DBRM library. The size of this dataset and the number of directory entries depend on the particular site (5 tracks and 20 directory blocks should be adequate for most environments). The PDS should have a fixed-block record format and a record length of 80.

Any standard dataset name can be used for this DBRM library; however, this installation procedure assumes that the name SAGLIB.SMADBRM is used.

Step 2: Generate the NDB 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 with INPL during the base Natural installation. NDBGENI contains the following two parameters, which you can modify to meet your specific requirements:

- the DB-environment parameter, which must be set to:
 - "DB2V23 " if you are running DB2 Version 3.1.,
 - "DB2V3" if you are running DB2 Version 3
 - "DB2V4" if you are running DB2 Version 4 or above
 - "DB2V5" if you are running DB2 Version 5 or above
- the parameter to specify the number of statements for dynamic access.

NDBIOMO performs the dynamic access to DB2 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 3: Assemble and link NDBIOMO - Job I055, Step 1610

Precompile, assemble and link NDBIOMO.

Note:

The link-edit step receives a condition code of 4 because of unresolved references for DSNHLI. This is normal and can be ignored.

Step 4: Create the NDB plan - Job I055, Step 1630

If desired, change library names and plan name to meet site requirements.

Step 5: Modify, assemble and link the NDB parameter module Job I055, Steps 1640/1650 or 1660/1670 or 1675/1676

The NDB parameter module contains the macro NDBPRM with parameters specific to the Natural interface to DB2.

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 Parameter Module NDBPARM.

- When the file server is **not** to be used:
Execute the steps 1640 and 1650; the resulting parameter module is called NDBPARM.
- When the file server is to be used:
Execute the steps 1660 and 1670; the resulting parameter module is called NDBPARMF.
- When the file server uses the Software AG Editor buffer pool as storage medium:
Execute the steps 1675 and 1676; the resulting parameter module is called NDBPARME.

Step 6: Link-edit NATGWDB2 - Job I055, Step 1680

Link-edit the environment-independent NDB nucleus NATGWDB2. Verify that the INCLUDE cards refer to the corresponding DD names for the load libraries.

Step 7: Modify, assemble and link NATPARM

Adapt your Natural parameter module NATPARM by adding parameters specific to Natural for DB2 and reassemble NATPARM.

Step 8: Relink your Natural nucleus

Natural for DB2 basically consists of:

- An environment-independent nucleus, which can be shared by multiple environments and which is therefore LPA-eligible.
- Environment-dependent components, which must be linked to the appropriate Natural environment-dependent interface.

Modify the JCL used to link your Natural shared nucleus by adding the following INCLUDE card:

INCLUDE SMALIB(NATGWDB2)	Environment-independent NDB Nucleus from Step 6
--------------------------	---

Modify the JCL used to link your Natural environment-dependent nucleus by adding the following INCLUDE cards and the corresponding DD statements:

INCLUDE SMALIB(NDBPARM)	NDB Parameter Module created in Step 5
INCLUDE SMALIB(NDBIOMO)	NDB I/O Module created in Step 3
INCLUDE DSNLIB(DSNTIAR)	SQL Error Message Module
INCLUDE .xxxxxxx(yyyyyyyy)	Environment-dependent DB2 Interface (see below)

If you want to use the Natural File Server, include SMALIB(NDBPARMF) or SMALIB(NDBPARME) instead of SMALIB(NDBPARM); see also Step 5 above.

Depending on your environment(s), INCLUDE the appropriate environment-specific language interface yyyyyyy in the library .xxxxxxx as shown in the following table:

Interface	Library	Environment
DSNALI	DSNLIB	Under TSO and in batch mode without running under the control of the DSN command processor (that is, with CAF).
DSNRLI	DSNLIB	WLM stored procedure address space.
DSNELI	DSNLIB	Under TSO and in batch mode when running under the control of the DSN command processor.
DSNCLI	DSNLIB	Under CICS
DFSLI000	IMSLIB	Under IMS/TM (MPP and BMP) and in batch mode using the DB2 DL/I batch support (DSNMTV01).
NDBC000	NDBLIB	Under Com-plete.

Note:

If you want to use NDB in various environments (that is, with different TP monitors), you must 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 do not use a Natural shared nucleus, all modules must be included in the link-edit of the Natural nucleus.
 2. Remove NATGWDB2 from the link-edit of the Natural shared nucleus and run it as a separate module with the mandatory entry name NATGWDB2. You can modify the name of the module created in Step 6. 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 module is arbitrary. This way of link-editing only applies if an alternative parameter module (PARM profile parameter) is used.
- 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:

TP-Monitor	Requirement
CICS	The resulting module must be defined via a PPT entry or RDO. PPT entry: DFHPPT TYPE=ENTRY , PROGRAM= <i>module-name</i> , PGMLANG=ASSEMBLER
Com-plete	The resulting module must be defined as RESIDENTPAGE or reside in the LPA/(E)LPA.

Step 9: Load Natural objects into system file - Job I061, Step 1600

Before executing this step, change the CMWKF01 DD statement to point to the NDB*nnn*.INPL dataset.

In this step, the NDB system programs, maps and DDMs are loaded into the Natural system files. The INPL job loads objects into the libraries SYSDDM, SYSTEM and SYSDB2.

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



Warning:
 Ensure that your newly created SYSDB2 library contains all necessary Predict interface programs, which are loaded into SYSDB2 when installing Predict (see the relevant Predict documentation).

Step 10: Load Natural error messages into system file - Job I061, Step 1620

Before executing this step, change the CMWKF02 DD statement to point to the NDB*nnn*.ERRN dataset.

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

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

Step 11: Create an NDB server stub - Job I070, Steps 1604,1606,1608,1610,1612,1614

Create a server stub to execute DB2 stored procedures written in Natural. A Natural for DB2 server stub is an interface module between the DB2 database system and the Natural server. In order to execute DB2 stored procedures written in Natural, the server stub needs to be installed.

There are 3 types of server stubs:

1. The server stub (NDB31SRV, steps 1604 and 1606).
The server stub's purpose is to execute a stored procedure written in Natural. If the associated Natural server environment is not yet initiated within the DB2 stored procedure address space, the Natural server environment is started implicitly.
2. The start server stub (NDB31STR, steps 1608 and 1610)
The start server stub's purpose is to start the Natural server environment explicitly. When it is called, it initiates the Natural server, the associated server NDB31SRV.
3. The termination server stub (NDB31TRM, steps 1612 and 1614)
A termination server stub has the purpose to terminate the Natural server environment explicitly. When it is called, it terminates the associated server NDB31SRV.

The Natural for DB2 server stubs are generated from the NDBSTUB macro. 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 Natural for DB2 server stub.

The resulting load modules have to be placed into a steplib library of the JCL used to execute the stored procedure address space. Each stored procedure written in Natural has to be declared in a row of the SYSIBM.SYSPROCEDURES table of the DB2 catalog and the LOADMOD column has to contain the name of the generated Natural for DB2 server stub module (NDB31SRV).

Steps Specific to a CICS Environment

Ensure that your Natural/CICS thread size is large enough to contain the DB2SIZE; if you use the Natural Tools for DB2, an additional storage of 8 KB is required.

This section covers the following topics:

- Using Plan Selection by CICS RCT Entry Threads
- Using Plan Selection by Dynamic Plan Exit
- Using the File Server with VSAM

Using Plan Selection by CICS RCT Entry Threads

If you want fixed assignment of your transaction code to the DB2 plan, add an additional entry to your RCT, or create a DB2Entry with RDO if you have CICS TS installed.

Step 1: Modify, assemble and link the CICS RCT or create a DB2Entry

Modify your RCT as follows (for any other parameters, refer to the relevant DB2 literature):

```
DSNRCT TYPE=ENTRY,PLAN=plan-name,TXID=(transaction-ID)
```

Or:

Define a DB2Entry with RDO (for parameters, refer to the relevant CICS literature):

```
DEFINE DB2ENTRY
OVERTYPE TO MODIFY                                CICS RELEASE = 0530
CEDA DEFINE DB2Entry(                               )
  DB2Entry      : DB25ENTR
  Group         : NCI23
  Description    :
THREAD SELECTION ATTRIBUTES
  TRansid       : transaction-id
THREAD OPERATION ATTRIBUTES
  ACcountrec    : None                               None ! TXid ! TAsk ! Uow
  AUTHid        :
  AUTHType      : Userid                             Userid ! Opid ! Group ! Sign ! TErM
                                                       ! TX
  DRollback     : Yes                               Yes ! No
  PLAN          : plan-name
  PLANExitname  :
  PRiority      : High                             High ! Equal ! Low
  PROtectnum    : 0005                              0-2000
  THREADLimit   : 0005                              0-2000
  THREADWait    : Pool                              Pool ! Yes ! No
```

The *plan-name* must be the same as the name used in step 4 (Create the NDB plan).

Using Plan Selection by Dynamic Plan Exit

If you want to perform plan selection by using the dynamic plan exit, perform the following steps:

Step 1: Assemble the CICS plan specification module NDBPLNA - Job I070, Step 1620

Precompile, assemble and link NDBPLNA for CICS.

Note:

This step receives a condition code of 4 because of an unresolved external reference for DFHEAI0 and DFHEI1. This is normal and can be ignored.

Step 2: Assemble the CICS dynamic plan selection exit module NDBUEXT - Job I070, Step 1630

The sample exit NDBUEXT can be modified to use a default plan name if none has been specified prior to the first SQL call. Review the source code in the module NDBUEXT for details about specifying a default plan name.

Optionally modify the source module NDBUEXT.

Precompile, assemble and link NDBUEXT for CICS.

Note:

This step receives a condition code of 4 because of an unresolved external reference for DFHEAI0 and DFHEI1. This is normal and can be ignored.

Step 3: Link-edit the CICS plan specification module NDBPLNA - Job I075, Step 1630

The resulting module NDBPLNA must be linked to the CICS load library and defined via a corresponding PPT entry or RDO.

PPT entry:

```
DFHPPT TYPE=ENTRY , PROGRAM=NDBPLNA , PGMLANG=ASSEMBLER
```

Step 4: Link-edit the CICS dynamic plan selection exit module NDBUEXT - Job I075, Step 1640

The resulting module NDBUEXT must be linked to the CICS load library and defined via a corresponding PPT entry or RDO.

PPT entry:

```
DFHPPT TYPE=ENTRY , PROGRAM=NDBUEXT , PGMLANG=ASSEMBLER
```

Step 5: Modify, assemble and link the CICS RCT or create a DB2Entry

Modify your RCT as follows (for any other parameters, refer to the relevant DB2 literature):

```
DSNRCT TYPE=POOL,PLNPGME=NDBUEXT,PLNEXIT=YES
```

The parameter PLNPGME must specify the same program as the NAME statement of the link step above.

Or:

Define a DB2Entry with RDO if you have CICS TS installed (for parameters, refer to the relevant CICS literature):

```
DEFINE DB2ENTRY
OVERTYPE TO MODIFY                                CICS RELEASE = 0530
CEDA DEFINE DB2Entry(                               )
  DB2Entry      : DB25ENTR
  Group         : NCI23
  Description    :
THREAD SELECTION ATTRIBUTES
  TRansid       : transaction-id
THREAD OPERATION ATTRIBUTES
  ACcountrec    : None                            None ! TXid ! TAsk ! Uow
  AUTHid        :
  AUTHType      : Userid                          Userid ! Opid ! Group ! Sign ! TTerm
                                                    ! TX
  DRollback     : Yes                             Yes ! No
  PLAN          :
  PLANExitname  : NDBUEXT
  PRiority      : High                            High ! Equal ! Low
  PROtectnum    : 0005                             0-2000
  THREADLimit   : 0005                             0-2000
  THREADWait    : Pool                            Pool ! Yes ! No
```

The parameter PLANExitname must specify the same program as the NAME statement of the link step above.

Alternatively or additionally, you can specify the plan exit program NDBUEXT with the PLANExitname parameter of POOL THREAD ATTRIBUTES of the DB2Conn resource definition of CICS TS.

Using the File Server with VSAM

If you want to use the Natural File Server (VSAM), perform the following additional steps:

Step 1: Define a VSAM dataset for the file server - Job I008, Step 1610

Specify the size and the name of the VSAM RRDS that is to be used as the file server.

Step 2: Format the file server dataset - Job I075, Step 1610

Specify the five input parameters required to format the file server dataset.

Step 3: Modify, assemble and link the CICS tables

Shown below are sample additional CICS table entries needed for the file server and for the DB2 components of Natural:

FCT entry:

```

CMFSERV  DFHFCT  TYPE=DATASET ,           X
          ACCMETH=VSAM ,                 X
          BUFND=5 ,                      X
          BUFNI=4 ,                      X
          DATASET=CMFSERV ,              X
          DISP=SHR ,                     X
          DSNAME=SAGLIB.NCIDB2.SERVER ,  X
          FILSTAT=( ENABLED ,CLOSED ) ,  X
          JID=NO ,                       X
          LOG=NO ,                       X
          LSRPOOL=NONE , 1-8 ONLY FOR XA; NONE X
          RECFORM=( FIXED ,BLOCKED ) ,   X
          RSL=PUBLIC ,                   X
          SERVREQ=( ADD ,UPDATE ,DELETE ,BROWSE ) , X
          STRNO=4

```

Step 4: Restart CICS

Restarting CICS is required, because of the additional FCT entry above.

Steps Specific to a Com-plete Environment

Under Com-plete, the installation procedure of NDB continues with the adaptation of your Com-plete environment.

Ensure that the changes required for DB2 have been applied to your Com-plete environment (see the relevant Com-plete documentation).

Steps Specific to an IMS/TM Environment

Ensure that your Natural/IMS roll buffer is large enough to contain the DB2SIZE; if you use the Natural Tools for DB2, an additional storage of 8 KB is required.

Bind Default NDB Plans for the Different IMS/TM Environments

- JOB I055 / Steps 1631, 1632, 1633, 1634 for IMS MPP conversational, IMS BMP Natural, IMS MPP non-conversational, OBMP Natural

If desired, change library names and plan names to meet site requirements.

Using Plan Selection by Using an IMS Resource Translation Table

If the name (or any ALIAS) of your environment-dependent Natural nucleus does not match the name of your DB2 plan, you must use an Resource Translation Table (RTT).

Step 1: Modify, assemble and link the IMS RTT

Add an additional DSNMAPN macro to your RTT as follows (for any other parameters, refer to the relevant DB2 literature):

DSNMAPN macro:

```
DSNMAPN APN=load-module, PLAN=plan-name
```

The *load-module* represents the environment-dependent Natural nucleus (that is, the IMS application program) and the *plan-name* is the same as the one used in the BIND step.

Using the File Server with VSAM

Be aware that database loops cannot be continued across terminal I/Os without using the File Server.

If you want to use the Natural File Server (VSAM), perform the following additional steps:

Step 1: Define a VSAM dataset for the file server - Job I008, Step 1600

Specify the size and the name of the VSAM RRDS that is to be used as the file server.

Step 2: Format the file server dataset - Job I075, Step 1600

Specify the five input parameters required to format the file server dataset.

Step 3: Update the JCL for the MPP region

- Include the DD statement CMFSERV to define the file server dataset.
- Increase the REGION parameter if necessary.

Step 4: Restart your Natural/IMS MPP region

Restart your MPP region, because of the additional DD statement.

Steps Specific to a TSO Environment

Using the File Server with VSAM

If you want to use the Natural File Server (VSAM), perform the following additional steps:

Step 1: Modify NDBFSRV in NATTSO

Set the NDBFSRV parameter in the NATTSO macro to "Yes" and reassemble and relink your Natural/TSO interface NATTSO.

Step 2: Define a VSAM dataset for the file server - Job I008, Step 1620

Specify the size and the name of the VSAM RRDS that is to be used as the file server.

Step 3: Format the file server dataset - Job I075, Step 1620

Specify the five input parameters required to format the file server dataset.

Sample JCL for Starting and Using Natural for DB2 under CAF

To test the TSO installation of NDB under CAF, perform the following steps:

Step 1: Adapt the clist NDBCAF - Job I070, Step 240C

Change the library and program names in the clist NDBCAF to meet site requirements. If you do not use the file server, remove the ALLOC and FREE statements for CMFSERV.

Step 2: Invoke Natural

Invoke Natural by executing the clist created in the previous step. Ensure that DB2 tables can be accessed and that plan switching can be performed.

Before the first SQL call you must call NATPLAN to explicitly allocate the plan. The plan name must be the same as the name used in step 4 (Create the NDB plan). NATPLAN can be edited to specify the appropriate DB2 subsystem ID.

Sample JCL for Starting and Using Natural for DB2 under DSN

To test the TSO installation of NDB under DSN, perform the following steps:

Step 1: Adapt the clist NDBTSO - Job I070, Step 240B

Change the subsystem ID as well as the library, plan and program names in the clist NDBTSO to meet site requirements. If you do not use the file server, remove the ALLOC and FREE statements for CMFSERV.

Step 2: Invoke Natural

Invoke Natural by executing the clist created in the previous step. Ensure that DB2 tables can be accessed. The plan name must be the same as the name used in the BIND step. For an explanation of the DSN and RUN commands, refer to the relevant IBM documentation for DB2 TSO and batch users.

Installation Verification

This section covers the following topics:

- Sample Batch Verification Jobs
- Online Verification Methods

Sample Batch Verification Jobs

To verify the installation of the Natural interface to DB2, several sample batch verification jobs are provided.

Test the Batch Mode Installation of NDB under CAF - Job NDBBATCA

NDBBATCA contains sample JCL to test NDB in batch mode using the CAF interface. Modify the sample JCL to meet site requirements.

Before the first SQL call you must call NATPLAN to explicitly allocate the plan. The plan name must be the same as the name used in step 4 (Create the NDB plan). NATPLAN can be edited to specify the appropriate DB2 subsystem ID.

Test the Batch Mode Installation of NDB under DSN - Job NDBBATTB

NDBBATTB contains sample JCL to test NDB in batch mode using the DSN command processor. Modify the sample JCL to meet site requirements.

The plan name must be the same as the name used in step 4 (Create the NDB plan). For an explanation of the DSN and RUN commands, refer to the relevant IBM documentation for DB2 TSO and batch users.

Test the DSNMTV01 Installation - Job NDBMTV01

NDBMTV01 contains a sample JCL to execute Natural using the DB2 DL/I batch support. Modify the sample JCL to meet site requirements.

The plan name must be the same as the name used in step 4 (Create the NDB plan).

Online Verification Methods

To verify the installation of the Natural interface to DB2 online, you can use either of the following two methods.

Using the SQL Services

► **To verify and check the installation of NDB using the SQL Services of the Natural SYSDDM utility**

1. Invoke Natural.
2. Invoke SYSDDM.
3. On the SYSDDM main menu enter function code "B" to invoke the SQL Services function.
Enter function code "S" and specify SQL system "DB2" to select all DB2 tables.
The communication between Natural and DB2 works if all existing DB2 tables are displayed.
For one of the tables, generate a Natural DDM as described.
4. After you have generated a DDM, access the corresponding DB2 table with a simple Natural program:

Example:

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

If you receive the message NAT3700, enter the Natural system command SQLERR to display the corresponding SQL return code. You can find the description of the SQLERR command in the section Natural System Commands for DB2 (Natural Tools for DB2).

Using the DEM2* Sample Programs

To verify and test your installation you can also use the sample programs DEM2* in the library SYSDB2 provided on the installation tape.

Using these sample programs, you can create a DB2 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.

Natural Parameter Modification for DB2

This section covers the following topics:

- Natural Profile Parameter Settings
- Performance Considerations for the DB2SIZE Parameter

Natural Profile Parameter Settings

▶ To set the Natural profile parameter

1. 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 DB2 buffer area, which should be set to at least 6 KB.

The setting of DB2SIZE also depends on whether you use the file server or not. If the file server is **not** used, the setting can be calculated according to the following formula:

$((808 + n1 * 40 + n2 * 80) + 1023) / 1024$ KB

If the file server is used, the setting can be calculated according to the following formula:

$((904 + n1 * 40 + n2 * 112 + n3 * 8) + 1023) / 1024$ KB

The variables *n1*, *n2* and *n3* correspond to:

<i>n1</i>	the number of statements for dynamic access as specified as the second parameter in Job I055, Step 1600;
<i>n2</i>	the maximum number of nested database loops as specified with the MAXLOOP parameter in NDBPARM;
<i>n3</i>	the maximum number of file server blocks to be allocated per user specified as the fifth parameter in Job I075, Step 1620 or the EBPMAX parameter of NDBPARM, if you decided to use the Software AG Editor buffer pool as file server.

Important:

Ensure that you have also added the Natural parameters required for the Software AG Editor; see the relevant installation description in the section Installing the Software AG Editor, in the Natural Installation Guide for Mainframes).

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

2. Add an NTDB entry specifying the list of logical database numbers that relate to DB2 tables. All Natural DDMs that refer to a DB2 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.

Important:

Ensure that all DB2 DDMs used when cataloging a given program have a valid DB2 DBID. Also ensure that the DBIDs selected in the NTDB macro for DB2 do not conflict with DBIDs selected for other database systems.

As of Natural Version 2.2, the DBID for DB2 used when cataloging a Natural program no longer has 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, since so far all DDMs have been cataloged with DBID 250.

Two sample NTDB macros follow:

NTDB DB2,250

NTDB DB2,(200,250,251)

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 DB2.

In previous Natural for DB2 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 DB2 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.

Depending on the SQL execution mode and on the usage of the Natural file server, the additional storage requirements (in bytes) can be calculated as follows:

- Dynamic Mode
- Static Mode
- Storage Requirements for the File Server
- Sample Calculation for Dynamic Mode without Using the File Server
- Considerations for VARCHAR Fields

Dynamic Mode

- 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.

Static Mode

- With sending fields:
 - $64 + n * 24$**
 - 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):
 - $64 + n * 24 + 24 + n * 2$**
 - where "n" is the number of receiving fields in an SQL statement.
 - The storage remains allocated until the loop is terminated.

Storage Requirements for the File Server

When using the file server, additional storage is required for each database loop that contains positioned UPDATE and/or DELETE statements.

For each of such loops, a buffer is allocated to save the contents of all receiving fields contained in the INTO list. Therefore, the size of this buffer corresponds to the total length of all receiving fields:

$$20 + \text{sum} (\text{length} (v1), \dots, \text{length} (vn))$$

where "v1 ... vn" refers to the variables contained in the INTO list.

The buffer remains allocated until the loop is terminated.

Sample Calculation for Dynamic Mode without Using the File Server

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 instead, 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.

Since 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.

Considerations for VARCHAR Fields

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.

Parameter Module NDBPARAM

The source module NDBPARAM contains Natural parameters specific to a DB2 environment. The parameter default values can be modified to meet site-specific requirements (see Step 5 of the Installation Procedure).

NDBPARAM 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 without the file server only).
DDFSERV	Alternate DD name for the file server.
DLISYNC	Obsolete and no longer documented. Only maintained for compatibility reasons.
EBPFSRV	Indicates that the Software AG Editor buffer pool is used as storage medium of the Natural for DB2 file server.
EBPPRAL	Specifies the number of blocks to be defined as primary allocation per user if the Software AG Editor buffer pool is to be used as file server.
EBPSEC	Specifies the number of blocks to be defined as secondary allocation per user if the Software AG Editor buffer pool is to be used as file server.
EBPMAX	Specifies the number of blocks to be defined as maximum allocation per user if the Software AG Editor buffer pool is to be used as file server.
ETIGN	Ignores END TRANSACTION errors under IMS/TM and continues processing.
FSERV	Use and mode of the file server.
MAXLOOP	Specifies the maximum number of nested program loops.
NNPSF	Set NATURAL NUMERIC's positive sign to F.
REFRESH	Allows the automatic setting of the DB2 server and package set at the beginning of a new database transaction to the values which were in place during the last database transaction.
STATDYN	Allows dynamic execution of statically generated SQL statements if the static execution returns an error.

The individual parameters are described in the following section. Their values cannot be dynamically overwritten.

Parameters in NDBPARM

BTIGN - Ignore Error after late BACKOUT TRANSACTION

Possible values:	Default value:
ON/OFF	ON

This parameter is relevant in CICS and IMS/TM environments only.

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

Value	Explanation
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

Possible values:	Default value:
ON/OFF	ON

This parameter is used to allow conversational mode in CICS environments where no file server is used.

Value	Explanation
ON	Conversational mode is allowed.
OFF	Conversational mode is not allowed.

If this parameter is set to OFF and no file server is used, you cannot continue database loops across terminal I/Os; if so, the DB2 SQL codes -501, 504, 507, 514, or 518 may occur.

If you are using the SYSDDM SQL services in a CICS environment without file server, you must specify CONVERS=ON, otherwise you get the errors mentioned above.

DDFSERV - Variable DD Name for CMFSERV

Possible values:	Default value:
Any valid DD name	None

This parameter specifies a DD name for the file server module other than "CMFSERV".

EBPFSRV

Possible values:	Default value:
ON/OFF	ON

This parameter is used to determine whether the file server uses the Software AG Editor buffer pool as storage medium or not. If it is OFF, the file server uses the VSAM file as in previous versions. The parameter must be set to ON, if the file server is to be used in a Sysplex environment.

EBPPRAL

Possible values:	Default value:
0 - 32676	20

This parameter specifies the number of blocks to be allocated primarily to each user of the file server, if the Software AG Editor buffer pool is used as storage medium.

If the EBPFSRV parameter is set to OFF, EBPPRAL is not used at runtime.

EBPSEC

Possible values:	Default value:
0 - 32676	10

This parameter specifies the number of blocks to be allocated secondarily to each user of the file server if the Software AG Editor buffer pool is used as storage medium. The secondary allocation is used to allocate buffer pool block to the user if the primary allocation amount is already exhausted.

If the EBPFSRV parameter is set to OFF, EBPSEC is not used at runtime.

EBPMAX

Possible values:	Default value:
0 - 32676	100

This parameter specifies the maximum number of blocks to be allocated to each user of the file server if the Software AG Editor buffer pool is used as storage medium. This parameter serves as upper limit for the allocation of buffer pool blocks to a single user. If the EBPFSRV parameter is set to OFF EBPMAX is not used at runtime.

ETIGN - Ignore END TRANSACTION Error under IMS/TM

Possible values:	Default value:
ON/OFF	ON

This parameter is relevant in IMS/TM MPP and message-oriented BMP environments only.

It is used to handle END TRANSACTION statements in a message-driven IMS region (MPP or message-oriented BMP).

In such a region, an END TRANSACTION cannot be executed by the Natural/IMS interface and is therefore ignored without any notification. In such situations, the ETIGN parameter can be used to issue an error message instead.

Value	Explanation
ON	The END TRANSACTION error is ignored and processing is continued.
OFF	The END TRANSACTION error is not ignored.

FSERV - Use and Mode of the Natural File Server

Possible values:	Default value:
ON/OFF/DIS	OFF

This parameter determines whether the file server is to be used and whether it can be disabled in the case of an initialization error.

Value	Explanation
ON	The file server is to be used.
OFF	The file server is not to be used.
DIS	The file server is to be used but is to be disabled if it cannot be initialized.

If FSERV is set to ON and the file server is not operational, the initialization of Natural for DB2 is terminated with a corresponding Natural error message. The Natural interface to DB2 is disabled and any SQL call is rejected with a corresponding error message.

MAXLOOP - Maximum Number of Nested Database Loops

Possible values:	Default value:
1 - 99	10

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

NNPSF - Set Natural NUMERIC's Positive Sign to F

Possible values:	Default value:
ON/OFF	OFF

This parameter changes the sign character of positive Natural variables which have format N, if they are filled from DB2. Usually these variables have the C as positive sign character. If the parameter NNPSF is set to ON, F is used as positive sign character.

Value	Explanation
ON	Positive numbers put into Natural NUMERIC variables by DB2 get the sign F.
OFF	Positive numbers put into Natural NUMERIC variables by DB2 get the sign as it was delivered for DB2.

REFRESH - Refresh DB2 Server and Package Set Automatically

Possible values:	Default value:
ON/OFF	OFF

This parameter is used to automatically set the DB2 server and package set to the values which were in place when the last transaction was executed. Server and package set are refreshed by using the `CONNECT TO server-name` and `SET CURRENT PACKAGESET = 'package-name'` SQL statements of DB2.

Value	Explanation
OFF	No automatic refreshment is performed.
ON	The automatic refreshment is done everytime before a database transaction starts and "old" values of server and package set are present.

STATDYN - Allow Static to Dynamic Switch

Possible values:	Default value:
NEVER/ALWAYS/SPECIAL	NEVER

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

Value	Explanation
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: DBRM does not exist in plan SQL -818: Mismatch of timestamps

Natural for DB2 Server Stub

The Natural server stub module sets up and invokes a Natural server in order to execute DB2 stored procedures written in Natural. The Natural for DB2 server stub is created from the NDBSTUB macro. Its parameters are used to determine the Natural server environment used to execute DB2 stored procedures written in Natural.

NDBSTUB contains the following parameters:

Parameter	Function
THREADSIZE	Determines the size of the Natural threads used by the Natural server.
THREADNUMBER	Determines the number of threads to be used by the Natural server.
NATURAL	Name of the server front-end or server Natural to be loaded by the server stub if the external CMSTART is not resolved by the linkage editor.
SERVER	Server name used to create the Natural server environment.
MODE	Operation mode of server stub.
MAIN	IBM Language Environment MAIN program option.
GTRCID	GTRACE ID to be used.
GTRACE	Specifies whether the server stub executes GTRACE macro calls.

The individual parameters are described in the following section. Their values cannot be dynamically overwritten.

THREADSIZE - Determines the Size of the Natural Threads Used by the Server Natural

Possible values:	Default value:
Decimal number	512

THREADSIZE determines the size of the Natural threads used by the server Natural. The size is specified in units of kilobytes.

THREADNUMBER - Determines the Number of Threads to be Used by the Server Natural

Possible values:	Default value:
Decimal number	10

THREADNUMBER determines the number of Natural threads used by the server Natural. This number influences the number of parallelly active stored procedures in the Natural server.

Keep in mind that the number of concurrently active DB2 stored procedures is limited by DB2 parameter. Therefore it does not make sense to set THREADNUMBER greater than the DB2 limit.

NATURAL - Name of the Server Front-End or Server Natural

Possible values:	Default value:
Any valid load module name	NATBAT31

NATURAL denotes the name of the server front-end or server Natural load module which will be loaded by the server stub if the external CMSTART is not already resolved by the linkage editor during creation of the server stub. The named load module has to be present in any steplib of the stored procedure address space.

SERVER - Server Name Used to Create the Natural Server Environment

Possible values:	Default value:
Up to 5 characters	NDB31

Server names suffixed with the three characters SRV denote the names of the servers used by the server front-end in order to identify the server Natural . These names should be unique within one address space.

MODE - Operation Mode of Server Stub

Possible values:	Default value:
STR/SRV/TRM	SRV

MODE determines the operational mode of the generated server stub.

Value	Explanation
STR	Invoking this server stub sets up the Natural server environment.
SRV	Invoking this server stub invokes the associated stored procedure, which is executed like a Natural subprogram in the Natural server. If the server environment is not yet set up in the stored procedure address space, the server stub will implicitly set up the Natural server environment.
TRM	Invoking this server stub terminates the Natural server environment in the stored procedure address space.

MAIN - IBM Language Environment Main Program Option

Possible values:	Default value:
YES/NO	YES

MAIN determines whether the generated server stub module will be an IBM Language Environment main program or not. It directly influences the MAIN parameter of the CEEENTRY macro.

Value	Explanation
YES	The generated server stub operates as IBM Language Environment main program.
NO	The generated server stub operates as IBM Language Environment sub program.

GTRCID - GTRACE ID to be Used

Possible values:	Default value:
Decimal number from 0 to 1023	203

GTRCID specifies the event ID recorded with the trace data created by the server stub.

GTRACE - Specifies Whether the Server Stub Executes GTRACE Macro Calls

Possible values:	Default value:
ON/OFF	OFF

GTRACE specifies whether or not the server stub executes GTRACE macro calls for tracing purposes.

Value	Explanation
ON	The generated server stub executes GTRACE macros in order to document its processing.
OFF	The generated server stub does not execute GTRACE macros during its processing cycle.

CMPRMIN - DDNAME of the CMPRMIN Dataset

Possible values:	Default value:
8 character DDNAME	CMPRMIN

CMPRMIN specifies the DDNAME of the CMPRMIN dataset during startup to read the input PROFILE parameter for this server.

CMPRINT - DDNAME of the CMPRINT Dataset

Possible values:	Default value:
8 character DDNAME	CMPRINT

CMPRINT specifies the DDNAME of the CMPRINT dataset to which the primary report output is written. If an asterisk (*) is specified, a unique DDNAME Pnnnnnnn is built whenever a stored procedure is invoked.

CMTRACE - DDNAME of the CMTRACE Dataset

Possible values:	Default value:
8 character DDNAME	CMTRACE

CMTRACE specifies the DDNAME of the CMTRACE dataset to which the primary report output is written. If an asterisk (*) is specified, a unique DDNAME Pnnnnnnn is built whenever a stored procedure is invoked which makes it possible to store each output separately.

Special Requirements for Natural Tools for DB2

To be able to use the Natural Tools for DB2, consider the following requirements and recommendations.

- Retrieval and Explain Functions
- LISTSQL and Explain Functions

Retrieval and Explain Functions

In order to be independent of DB2 versions, the Natural Tools for DB2 Retrieval and Explain functions have been designed not to access the DB2 catalog tables directly, but to access identical tables qualified by the creator name SYSSAG.

Thus, before you can use the Retrieval or Explain functions, you must create these tables. The SYSSAG tables must have the same columns as the catalog tables of DB2 Version 4 and they must be created as ALIAS, VIEW, or TABLE.

To help you create these tables, sample SQL code is provided in the members DEMSQL3 and DEMSQL4 in the library SYSD2, where DEMSQL4 is intended for DB2 Version 4. By default, it creates an ALIAS SYSSAG.xxx for the corresponding SYSIBM table.

With DB2 Version 3 you must use DEMSQL3. For some tables this member contains CREATE VIEW statements which emulate the columns of the catalog of DB2 Version 4. It also contains CREATE TABLE statements for tables that are not part of the catalog of DB2 Version 3.

For some catalog tables no indexes are defined. For performance reasons, you should therefore consider creating copies of these tables with appropriate indexes.

For the following tables it is recommended to work with copies of the catalog tables:

```
SYSCOLAUTH  
SYSDBRM  
SYFOREIGNKEYS  
SYSINDEXPART  
SYSKEYS  
SYSSTMT  
SYSSYNONYMS  
SYSTABLEPART  
SYSVIEWS
```

The required CREATE TABLE and CREATE INDEX statements are included as comments in the sample SQL members DEMSQL3 and DEMSQL4. Also sample SQL to update the data in the copies of the catalog tables is included in the member DEMSQLUP.

For any other table, it is recommended that you create an ALIAS or a VIEW that points to the corresponding SYSIBM table.

Note:

The sample SQL members can be executed with the ISQL part of SYSD2. ISQL enables you to read SQL members from the library SYSD2. To save an SQL member in any other library, you can use the command LIBRARY MYLIB in the ISQL input screen to switch to another library and then save the SQL member. You cannot save SQL members in the library SYSD2.

LISTSQL and Explain Functions

These functions access DB2 PLAN_TABLEs. To use these functions, a PLAN_TABLE must exist for your SQLID. For the layout of the PLAN_TABLE, see the relevant DB2 documentation of the EXPLAIN command. The PLAN_TABLE must have either the layout of DB2 Version 3 (34 columns) or the layout of DB2 Version 4 (35 columns).

It is recommended that you create an index on the following columns of the PLAN_TABLE:

DB2 Version 3	DB2 Version 4
APPLNAME	APPLNAME
PROGNAME	PROGNAME
	COLLID
QUERYNO	QUERYNO
TIMESTAMP DESC	TIMESTAMP DESC
QBLOCKNO	QBLOCKNO
PLANNO	PLANNO
MIXOPSEQ	MIXOPSEQ