

Installing Natural for DB2

This section describes how to install Natural for DB2 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.

This section covers the following topics:

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

Notation vrs or vr: If used in the following document, the notation *vrs* or *vr* stands for the relevant version, release, system maintenance level numbers. For further information on product versions, see *Version* in the *Glossary*.

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 the section Installing Natural for DB2, 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 Natural for DB2 installation tape; you must adapt this example job to your requirements. Note that the job numbers on the tape are preceded by the product code "NDB" of Natural for DB2 (for example, NDBI070).

Using System Maintenance Aid

For information on the use of Software AG's System Maintenance Aid for the installation process, refer to the *System Maintenance Aid* documentation.

Prerequisites

- Base Natural must be installed first; you cannot install Natural and Natural for DB2 at the same time.
- The Software AG Editor must be installed (see *Installing the Software AG Editor* in the *Installation* documentation).

For special considerations on the various environments supported by Natural for DB2, see *Environment-Specific Considerations*.

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

Installation Tape for Natural for DB2

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
NDBvrs.SRCE	Natural for DB2 source modules
NDBvrs.LOAD	Natural for DB2 load modules
NDBvrs.INPL	Natural for DB2 utility programs in INPL format
NDBvrs.ERRN	Natural for DB2 error messages
NDBvrs.JOBS	Natural for DB2 installation jobs
NDBvrs.LDEL	Instructions to delete Natural for DB2 objects of Version 4.1

Copying the Tape Contents to a z/OS Disk

If you are using SMA, refer to the *System Maintenance Aid* documentation (included in the current edition of the Natural documentation CD).

If you are *not* using SMA, follow the instructions below.

This section explains how to:

- Copy dataset COPY.JOB from tape to disk.
- Modify this dataset to conform to your local naming conventions.

The JCL in this dataset is then used to copy all datasets 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 Dataset COPY.JOB from Tape to Disk
- Step 2 - Modify COPY.JOB on Your Disk
- Step 3 - Submit COPY.JOB

Step 1 - Copy Dataset COPY.JOB from Tape to Disk

The dataset COPY . JOB (Label 2) contains the JCL to unload all other existing datasets 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=tape-volume),
// LABEL=(2,SL)
//SYSUT2 DD DSN=hilev.COPY.JOB,
// DISP=(NEW,CATLG,DELETE),
// UNIT=3390,VOL=SER=volume,
// SPACE=(TRK,(1,1),RLSE),
// DCB=*.SYSUT1
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
//
```

where:

hilev is a valid high level qualifier

tape-volume is the tape volume name, for example: T12345

volume is the disk volume name

Step 2 - Modify COPY.JOB on Your Disk

Modify the COPY . JOB on your disk to conform to your local naming conventions and set the disk space parameters before submitting 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 datasets from the tape to your disk.

Installation Procedure for Natural for DB2

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

This section covers the following topics:

- Common Installation Steps
- Installation Steps Specific to CICS
- Using Plan Selection by Dynamic Plan Exit
- Using the File Server with VSAM
- Installation Steps Specific to Com-plete
- Installation Steps Specific to IMS TM
- Installation Steps Specific to TSO

Common Installation Steps

The following steps are required to install Natural for DB2 in all supported environments.

Step 1: Allocate the DBRM Library for Use with Natural for DB2

Allocate a PDS as database request module (DBRM) library. The size of this dataset and the number of directory entries depend on the particular site (5 tracks and 20 directory blocks must be adequate for most environments). The PDS must 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.DB2DBRM` is used.

Step 2: Generate the Natural for DB2 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 NDBGENTI, which is loaded with INPL during the base Natural installation. NDBGENTI contains the following two parameters, which you can modify to meet your specific requirements:

- the DB-environment parameter, which must be set to:
 - DB2V7 if you are running DB2 Version 7
 - DB2V8 if you are running DB2 Version 8
 - DB2V9 if you are running DB2 Version 9 or higher
- the number of parallel dynamic prepared DB2 statements.

NDBIOMO (see also the relevant section in *Internal Handling of Dynamic Statements*) 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. Check the report for successful job 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: Bind the DBRM NDBIOMO into a Package

Job I055, Step 1620

If desired, change library names to meet site requirements.

Step 5: Create the DB2 Plan for Use with Natural for DB2

Job I055, Step 1630

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

Step 6: Modify, Assemble and Link the Natural for DB2 Parameter Module

Job I055, Steps 1640/1650 or 1660/1670 or 1675/1676

The Natural for DB2 parameter module NDBPARM contains the macro NDBPRM with parameters specific to Natural for 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 additional parameter module is called NDBPARMF.
- **When the file server uses the Software AG Editor buffer pool as the storage medium:**
Execute the Steps 1675 and 1676; the resulting additional parameter module is called NDBPARME.

Step 7: Link-Edit NATGWDB2

Job I055, Step 1680

Link-edit the environment-independent Natural for DB2 nucleus NATGWDB2.

Verify that the INCLUDE cards refer to the corresponding DD names for the load libraries.

Step 8: Modify, Assemble and Link the Natural Parameter Module

Adapt your Natural parameter module NATPARM by adding parameters specific to Natural for DB2 (see *Natural Parameter Modification for DB2*) and reassemble NATPARM.

Step 9: Relink your Natural Nucleus

Natural for DB2 basically consists of:

- An environment-independent nucleus, which can be shared by multiple environments.
- 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 Natural for DB2 nucleus from <i>Step 7: Link-Edit NATGWDB2</i>
-----------------------------	--

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)	Natural for DB2 parameter module created in <i>Step 6: Modify, Assemble and Link the Natural for DB2 Parameter Module</i>
INCLUDE SMALIB(NDBIOMO)	Natural for DB2 I/O module created in <i>Step 3: Assemble and Link NDBIOMO</i>
INCLUDE DSNLIB(DSNTIAR)	SQL Error Message Module
INCLUDE xxxxxxx(yyyyyyy)	Environment-dependent DB2 interface (see below)

If you want to use the Natural File Server for DB2, include SMALIB(NDBPARMF) or SMALIB(NDBPARME) instead of SMALIB(NDBPARM); see also *Step 6: Modify, Assemble and Link the Natural for DB2 Parameter Module*.

Depending on your environment(s), INCLUDE the appropriate environment-specific language interface yyyyyyy in the library xxxxxxxx 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 (Workload Manager) stored procedure address space and Natural Development Server (NDV) (recommended). Also usable in TSO and batch environments.
DSNELI	DSNLIB	Under TSO and in batch mode when running under the control of the DSN command processor.
DSNCLI	DFHLIB	Under CICS
DFSLI000	IMSLIB	Under IMS TM (MPP and BMP) and in batch mode by using the DB2 DL/I batch support (DSNMTV01).
NDBCOM	NDBLIB	Under Com-plete.

Note:

If you want to use Natural for DB2 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 7: Link-Edit NATGWDB2*. 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 Natural for DB2 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 10: Delete Natural for DB2 Objects

Job I061, Step 0016

This step is optional but recommended to avoid data inconsistencies.

If you are using a Version 4.1 Natural FNAT system file, delete obsolete Version 4.1 Natural for DB2 objects by loading the `NDBvrs.LDEL` data set with the Natural INPL utility.

See also the corresponding step *Delete Natural System Objects* in *Installation Procedure for Natural under z/OS* in the *Installation* documentation.

Step 11: Load Natural for DB2 Objects into the System File

Job I061, Step 1610

Before executing this step, change the `CMWKF01` DD statement to point to the `NDBnnn.INPL` dataset.

In this step, the Natural for DB2 system programs, maps and DDMs are loaded into the Natural system files. The INPL job loads objects into the Natural system libraries `SYSDDM`, `SYSTEM` and `SYSDB2`.

The Natural for DB2 system programs *must* be loaded into the 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 12: Load Natural for DB2 Error Messages into the System File

Job I061, Step 1620

Before executing this step, change the `CMWKF02` DD statement to point to the `NDBnnn.ERRN` dataset.

This step executes a batch Natural job that runs an error load program by using the `NDBnnn.ERRN` dataset as input. The `ERRLODUS` job loads error messages into the library `SYSERR` in the FNAT system file.

The Natural for DB2 error messages *must* be loaded into the FNAT system file.

Step 13: Create the Natural for DB2 Server Stub

Job I070, Steps 1604,1606,1608,1610

Create server stubs to execute Natural stored procedures and Natural user-defined functions. Natural for DB2 server stubs are interface modules between the DB2 database system and the Natural server. In order to execute Natural stored procedures and Natural user-defined functions, the server stub needs to be installed.

There are two types of server stub (*vr* in the stub name stands for the current product version and release numbers):

1. Natural for DB2 server stub (module NDBvrSRV, Steps 1604 and 1606)

The server stub is used to execute Natural stored procedures and Natural user-defined functions.

The IBM LE (Language Environment) runtime modules required must be linked to the Natural for DB2 server stub module. Use the `CALL` option of the linkage editor and assign the LE runtime library as `SYSLIB`.

2. Natural for DB2 start server stub (module NDBvrSTR, Steps 1608 and 1610)

The start server stub is used to start the Natural server environment(s) explicitly.

The IBM LE (Language Environment) runtime modules required must be linked to the Natural for DB2 start server stub module. Use the `CALL` option of the linkage editor and assign the LE runtime library as `SYSLIB`. Additionally, include the modules `NDBSTRP` (delivered with Natural for DB2) and `NATCONFIG` (delivered with Natural) from `NDBnnn.LOAD` and `NATnnn.LOAD`.

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 DB2 stored procedure address space.

For DB2 UDB, each Natural stored procedure or Natural user-defined function must be defined by a DB2 `CREATE PROCEDURE` or DB2 `CREATE FUNCTION` statement, where the name of the Natural for DB2 server stub module `NDBvrSRV` generated is specified as `EXTERNAL NAME`.

Step 14: Bind the DBRM ROUTINEN into a Package

Job I070, Step 1615

Bind the DBRM `ROUTINEN` into a package. The DBRM `ROUTINEN` is contained in the collection `SAGNDBROUTINENPACK` and delivered with Natural for DB2. Natural for DB2 needs this collection for accessing the DB2 catalog and retrieving the parameter descriptions of Natural stored procedures and Natural user-defined functions.

Installation Steps Specific to CICS

This section describes how to install Natural for DB2 in 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.

Natural for DB2 uses the CICS DB2 attachment facility to access DB2 for z/OS from a CICS environment. This requires the DB2 plan containing the SQL statements performed from Natural for DB2 have to be defined in the CICS resource definition. For this purpose, CICS DB2 provides three types of resource definitions:

- **DB2CONN**
Defines the connection between CICS and DB2 for z/OS.
- **DB2ENTRY**
Define threads used for access from CICS to DB2 for z/OS and their associated necessary resources, such as the DB2 plan.

- **DB2TRAN**

Associate CICS transactions with a DB2ENTRY definition.

There are two ways to select the DB2 plan for the transaction used by Natural for DB2:

- **Fixed DB2 plan definition**

Fixed DB2 plan definition is achieved by defining the PLAN attribute as the DB2 plan for the DB2ENTRY which is associated with the CICS transaction used by Natural for DB2.

- **Dynamic DB2 plan exit**

Dynamic DB2 plan exit is a CICS program which is invoked by the CICS DB2 attachment facility to determine the plan to be used for the thread at execution time.

Dynamic DB2 plan exit is achieved by defining the PLANEXIT attribute as the DB2 plan exit program for the DB2ENTRY which is associated with the CICS transaction used by Natural for DB2. Natural for DB2 provides the NDBUEXT program as DB2 plan exit. NDBUEXT provides the plan to CICS DB2, which has been previously set by executing the NATPLAN program.

This section covers the following topics:

- Using Plan Selection by CICS RCT Entry Threads

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 define a DB2Entry with RDO.

Perform one of the following alternative steps:

- **Modify, assemble and link CICS RCT**

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

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

- **Define a DB2Entry with RDO**

For parameters, refer to the relevant CICS literature by IBM.

```
DEFINE DB2ENTRY
OVERTYPE TO MODIFY                                CICS RELEASE = 0650
CEDA DEFINE DB2Entry(                               )
  DB2Entry      : DB2ENTR
  Group         : NCI
  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         : 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 to create the DB2 plan for Natural for DB2: see *Common Installation Steps*.

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 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 2: 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 3: Modify, Assemble and Link the CICS RCT or Define a DB2Entry

Perform one of the following alternative steps:

- **Modify your RCT**

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

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

The parameter PLNPGME must specify the same program as the NAME statement of *Step 2: Link-Edit the CICS Dynamic Plan Selection Exit Module NDBUEXT*.

- **Define a DB2Entry with RDO**

For parameters, refer to the relevant CICS literature by IBM.

```

DEFINE DB2ENTRY
OVERTYPE TO MODIFY                                CICS RELEASE = 0530
CEDA DEFine DB2Entry(                               )
  DB2Entry      : DB2ENTR
  Group         : NCI
  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 *Step 2: Link-Edit the CICS Dynamic Plan Selection Exit Module NDBUEXT*.

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 with VSAM system files, 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 (see also *Preparations for Using the File Server in Natural File Server for DB2*).

Step 2: Format the File Server Dataset

Job I075, Step 1610

Specify the five input parameters required to format the file server dataset (see also *Preparations for Using the File Server in Natural File Server for DB2*).

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
          DSNNAME=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.

Installation Steps Specific to Com-plete

Under Com-plete, the installation procedure of Natural for DB2 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).

Installation Steps Specific to IMS TM

This section describes how to install Natural for DB2 in an IMS TM environment.

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

Below is information on:

- Bind Default DB2 Plans for Different IMS TM Environments
- Using Plan Selection with IMS TM Resource Translation Table
- Using the File Server with VSAM

Bind Default DB2 Plans for Different IMS TM Environments

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

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

Using Plan Selection with IMS TM 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).

Below is information on:

- **Modify, assemble and link the IMS TM Resource Translation Table**

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

DSNMAPN macro:

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

The *load-module* represents the environment-dependent Natural nucleus (that is, the IMS TM 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 with VSAM system files, perform the following additional steps:

1. **Define the 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 (see also *Preparations for Using the File Server in Natural File Server for DB2*).

2. **Format the file server dataset**

Job I075, Step 1600

Specify the five input parameters required to format the file server dataset (see also *Preparations for Using the File Server in Natural File Server for DB2*).

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.

4. **Restart the MPP region used by your Natural IMS TM Interface**

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

Installation Steps Specific to TSO

This section describes how to install Natural for DB2 in a TSO environment:

- Using the File Server with VSAM
- Sample JCL for Starting and Using Natural for DB2 under CAF
- Sample JCL for Starting and Using Natural for DB2 under DSN

Using the File Server with VSAM

If you want to use the Natural File Server with VSAM system files, perform the following additional steps:

1. Modify NDBFSRV in NATTSO

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

2. Define the 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 (see also *Preparations for Using the File Server in Natural File Server for DB2*).

3. Format the file server dataset

Job I075, Step 1620

Specify the five input parameters required to format the file server dataset (see also *Preparations for Using the File Server in Natural File Server for DB2*).

Sample JCL for Starting and Using Natural for DB2 under CAF

To test the TSO installation of Natural for DB2 under CAF, perform the following steps:

1. Adapt 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.

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 5: Create the DB2 Plan for Use with Natural for DB2*. 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 Natural for DB2 under DSN, perform the following steps:

1. Adapt 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.

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 literature for DB2 TSO and batch users.

Installation Verification

This section provides example batch jobs and online methods for verifying the installation of Natural for DB2:

- Test Batch Natural for DB2 under CAF - Job NDBBATCA
- Test Batch Natural for DB2 under DSN - Job NDBBATTB
- Test DSNMTV01 - Job NDBMTV01
- Online Verification Methods

Test Batch Natural for DB2 under CAF - Job NDBBATCA

NDBBATCA contains sample JCL to test Natural for DB2 in batch mode by using the Call Attachment Facility (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 5: Create the DB2 Plan for Use with Natural for DB2*. NATPLAN can be edited to specify the appropriate DB2 subsystem ID.

Test Batch Natural for DB2 under DSN - Job NDBBATTB

NDBBATTB contains sample JCL to test Natural for DB2 in batch mode by 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 5: Create the DB2 Plan for Use with Natural for DB2*. For an explanation of the DSN and RUN commands, refer to the relevant IBM literature for DB2 TSO and batch users.

Test DSNMTV01 - Job NDBMTV01

NDBMTV01 contains a sample JCL to execute Natural by 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 5: Create the DB2 Plan for Use with Natural for DB2*.

Online Verification Methods

To verify the installation of Natural for DB2 online, you can use either SQL Services or DEM2 example programs:

- Using SQL Services
- Using DEM2* Example Programs

Using SQL Services

 **To verify and check the Natural for DB2 installation by 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 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 in the section *Generate DDM from an SQL Table in Generating Natural Data Definition Modules (DDMs)*.

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. .

Using DEM2* Example Programs

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

Using these example programs, you can create a DB2 table by using DEM2CREA and create the corresponding DDM using the Natural SYSDDM utility.

You can then store data in the created table by using DEM2STOR and retrieve data from the table by using DEM2FIND or DEM2SEL.

You can also drop the table by 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 must 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:

$$((1064 + n1 * 40 + n2 * 120) + 1023) / 1024 \text{ KB}$$

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

$$((1160 + n1 * 40 + n2 * 160 + n3 * 8) + 1023) / 1024 \text{ KB}$$

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* of the Natural *Installation* documentation.

As DB2SIZE applies to Natural for DB2 and Natural for SQL/DS, it must 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.

The DBID for SQL/DS used when cataloging a Natural program does not have to be in the NTDB list of DBIDs used when executing this program. Therefore, when executing existing Natural programs, DBID 250 is not mandatory. 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; see *Natural Profile Parameter Settings*.

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:

$$80 + n * 56$$

With sending fields including LOB columns:

$$80 + 2 * n * 56$$

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):

$$80 + n * 56 + 24 + n * 2$$

With receiving fields including LOB columns:

$$80 + 2 * n * 56 + 24 + n * 2$$

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:

$$80 + n * 24$$

With sending fields including LOB columns:

$$80 + 2 * n * 56$$

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):

$$80 + n * 24 + 24 + n * 2$$

With receiving fields including LOB columns:

$$80 + 2 * n * 56 + 24 + n * 2$$

where n is the number of receiving fields in an SQL statement.

The storage remains allocated until the loop is terminated.

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:

$$20 + 4 + \text{sum}(\text{length}(v1), \dots, \text{length}(vn))$$

where $v_1 \dots v_n$ 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 (n_1 and n_2), the calculated DB2SIZE will be 2208 bytes. However, if you specify a DB2SIZE of 20 KB instead, the available space for dynamically allocated storage will be 18272 bytes, which means enough space for up to either 325 sending fields or 313 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 110 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 NDBPARM

The source module NDBPARM is used in several Natural add-on products. It contains parameter macros specific to an SQL environment:

- NDBPRM
- NDBID

These macros are described below.

Parameter Macro NDBPRM

The default values of the parameters contained in this macro can be modified to meet site-specific requirements (see the corresponding step in *Common Installation Steps*). The values of the parameters cannot be dynamically overwritten.

Below is a description of all parameters contained in the NDBPRM macro:

BTIGN | CONVERS | CONVR2 | DDFSERV | DELIMID | EBPFSRV | EBPPRAL | EBPSEC |
 EBPMAX | ETIGN | FSERV | MAXLOOP | NNPSF | PSCIGN | REFRESH | RETRYPO | RWRDONL |
 STATDYN

BTIGN - Ignore BACKOUT TRANSACTION Error

This parameter enables you to ignore the error which results from a BACKOUT TRANSACTION statement that was issued too late for backing out the current transaction, because an implicit Syncpoint has previously been issued by the TP monitor.

Possible Values:

Value	Explanation
ON	The error after a late BACKOUT TRANSACTION is ignored. This is the default value.
OFF	The error after a late BACKOUT TRANSACTION is <i>not</i> ignored.

CONVERS - Conversational Mode under CICS

This parameter is used to enabled conversational mode in CICS environments where no Natural for DB2 file server is used.

Possible Values:

Value	Explanation
ON	Conversational mode is allowed. This is the default value.
OFF	Conversational mode is <i>not</i> allowed.

If this parameter is set to OFF and no Natural 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 Natural for DB2 file server, you must specify CONVERS=ON, otherwise you get the errors mentioned above.

CONVRS2 - Allow Conversational Mode 2 under CICS

This parameter is used to allow conversational mode 2 in CICS environments.

Possible Values:

Value	Explanation
ON	Conversational mode 2 is allowed.
OFF	Conversational mode 2 is <i>not</i> allowed. This is the default value.

This parameter is used to control conversational mode 2 in CICS environments. Conversational mode 2 means that update transactions are spawned across terminal I/Os until either an explicit COMMIT or explicit ROLLBACK has been issued (Caution: DB2 and CICS resources are kept across terminal I/Os!). This means CONVRS2=ON has the same effect as the Natural parameter PSEUDO=OFF, except that the conversational mode is entered after an DB2 update statement (UPDATE, DELETE, INSERT) and left again after a COMMIT or ROLLBACK, while PSEUDO=OFF causes conversational mode for the total Natural session.

See also CALLNAT subprogram NDBCONV, which allows setting or resetting conversational mode 2 dynamically.

DDFSERV - Alternate DD Name for Natural File Server

This parameter specifies a DD name for the Natural for DB2 file server module other than CMFSERV.

Possible Values:

Value	Explanation
<i>DD-name</i>	Any valid DD name. There is no default value.

DELIMID - Escape Character for Delimited Identifiers

This parameter determines the escape character to be used for generating delimited SQL identifiers for the column names and table names in SQL statements. A delimited identifier is a sequence of one or more characters enclosed in escape characters. You must specify a delimited identifier if you use SQL-reserved words for column names and table names, as demonstrated in the *Example of DELIMID* below.

Possible Values:

Value	Explanation
"	Double quotation mark
'	Single quotation mark
None	No value: Delimited identifiers are not enabled. This is the default value.

To enable generation of delimited identifiers, DELIMID must be set to double quotation mark (") or single quotation mark (').

The escape character specified for DELIMID and the SQL STRING DELIMITER are mutually exclusive. This implies that the mark (double or single quotation) used to enclose alphanumeric strings in SQL statements must be different from the value specified for DELIMID. If you enable delimited identifiers, ensure that the value specified for DELIMID also complies with the SQL STRING DELIMITER value of your DB2 installation.

See also the RWRDONL parameter to determine which delimited identifiers are generated in the SQL string.

Example of DELIMID:

In the following example, a double quotation mark (") has been specified as the escape character for the delimited identifier:

Natural statement:

```
SELECT FUNCTION INTO #FUNCTION FROM XYZ-T1000
```

Generated SQL string:

```
SELECT "FUNCTION" FROM XYZ.T1000
```

EBPFSRV - Editor Buffer Pool for Natural File Server

This parameter is used to determine whether the Natural file server uses the Software AG Editor buffer pool as the storage medium.

Possible Values:

Value	Explanation
ON	The Software AG buffer pool is to be used as the storage medium for the Natural file server. ON <i>must</i> be set if the file server is to be used in a Parallel Sysplex environment. In this case, your Natural session must use the auxiliary editor buffer pool (see also <i>Support of a z/OS Parallel Sysplex Environment</i> in the <i>Installation</i> documentation).
OFF	A VSAM file is to be used as the storage medium for the Natural file server. This is the default value.

EBPPRAL - Editor Buffer Pool Primary Allocation

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

Possible Values:

Value	Explanation
0 - 32676	Number of blocks to be allocated primarily.
20	This is the default value.

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

EBPSEC - Editor Buffer Pool Secondary Allocation

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

Possible Values:

Value	Explanation
0 - 32676	Number of blocks to be allocated secondarily.
10	This is the default value.

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

EBPMAX - Editor Buffer Pool Maximum Allocation

This parameter specifies the maximum number of blocks to be allocated to each user of the Natural file server if the Software AG Editor buffer pool is used as the storage medium. This parameter serves as upper limit for the allocation of buffer pool blocks to a single user.

Possible Values:

Value	Explanation
0 - 32676	Maximum number of blocks to be allocated.
100	This is the default value.

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

ETIGN - Ignore END TRANSACTION Error

This parameter is relevant in IMS 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 TM Interface and is therefore ignored without any notification. In such situations, the ETIGN parameter can be used to issue an error message instead.

Possible Values:

Value	Explanation
ON	The END TRANSACTION error is ignored and processing is continued. This is the default value.
OFF	The END TRANSACTION error is <i>not</i> ignored.

FSERV - Activate Natural File Server

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

Possible Values:

Value	Explanation
ON	Natural file server is to be used.
OFF	Natural file server is not to be used. This is the default value.
DIS	Natural file server is to be used but is to be disabled if it cannot be initialized.

If `FSERV` is set to `ON` and the Natural 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 Program Loops

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

Possible Values:

Value	Explanation
1 - 99	Maximum possible number of nested database loops.
10	This is the default value.

NNPSF - Set Natural Numerics' Positive Sign to F

This parameter changes the sign character of positive Natural variables which have format `N`, if they are filled from the SQL database system. 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.

Possible Values:

Value	Explanation
ON	Positive numbers put into Natural numeric variables by the SQL database system get the sign <code>F</code> .
OFF	Positive numbers put into Natural numeric variables by the SQL database system remain unchanged. This is the default value.

PSCIGN - Treat Positive Sqlcodes as Sqlcode 0

This parameter influences the treatment of positive sqlcodes returned from the SQL database system. If the parameter `PSCIGN` is set to `OFF`, a `NAT3700` error message is issued. If the parameter `PSCIGN` is set to `ON`, positive sqlcodes are treated as if they were zero, that is, no `NAT3700` error message is issued.

Possible Values:

Value	Explanation
ON	Positive sqlcodes are treated as zero.
OFF	Positive sqlcodes cause a <code>NAT3700</code> error message. This is the default value.

REFRESH - Refresh Setting of DB2 Server and Package Set

This parameter is used to automatically set the DB2 server and package set to the values that applied 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.

Possible Values:

Value	Explanation
ON	An automatic refresh is performed every time before a database transaction starts and if a server or package set has been specified.
OFF	No automatic refresh is performed. This is the default value.

RETRYPO - Number of Positioning Retries

This parameter delimits the number of retries done by Natural for DB2 in order to reposition a dynamic scrollable cursor in a pseudo-conversational environment (IMS MPP or CICS).

Possible Values:

Value	Explanation
0 - 2147483648	Number of retries done by Natural for DB2.
10	This is the default value.

This parameter applies only for dynamic scrollable cursors.

In pseudo-conversational environments, cursors are closed at terminal I/O. For dynamic scrollable cursors the current absolute position number and the current key column values are saved. After terminal I/O the dynamic scrollable cursor is opened again and positioned absolutely to the position of the saved absolute position. The contents of the key columns are compared with the saved values. If they match, processing continues with the next requested database operation.

If the contents of the key columns do not match the saved values, the next rows are fetched and compared with the saved values until either the values match or no row is found or the RETRYPO count is exhausted. In the latter cases the cursor is repositioned to the saved position and the prior rows are fetched and compared until either the values match or no row is found or the RETRYPO count is exhausted. In the latter cases a NAT3703 error message is issued. If a row is fetched whose key columns matches the saved values, processing continues with the next database instruction.

RETRYPO delimits the retries in each direction (*next* or *prior*).

If RETRYPO is zero no repositioning takes place.

RWRDONL - Generate Delimited Identifiers for Reserved Words Only

This parameter determines which identifiers are generated as delimited identifier in an SQL string. RWRDONL only takes effect if the setting of the DELIMID parameter allows delimited identifiers.

Possible Values:

Value	Explanation
ON	Only identifiers that are reserved words are generated as delimited identifiers. The list of reserved words is contained in the NDBPARM macro. This list has been merged from the lists of reserved words for DB2 for z/OS, DB2 for VSE/VM, DB2 for LINUX, OS/2, Windows and UNIX, and ISO/ANSI SQL99. This is the default value.
OFF	All identifiers are generated as delimited identifiers.

STATDYN - Allow Static to Dynamic Switch

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

Possible Values:

Value	Explanation
NEVER	Dynamic execution is never allowed. This is the default value.
ALWAYS	Dynamic execution is always allowed after an error.
SPECIAL	Dynamic execution is allowed after special errors only. These special errors are: <ul style="list-style-type: none"> ● NAT3706: Load module not found ● SQL -805: DBRM (database request module) does not exist in plan ● SQL -818: Mismatch of timestamps

Parameter Macro NDBID

The parameter macro NDBID determines the database type of an SQL DBID.

The NDBID macro is specified as follows:

1. Default Database Definition

The default database type is specified as follows. It applies to all database IDs not explicitly specified by NDBID.

```
NDBID=database-type
```

2. Single Database Definition

A single database ID and its type is specified as follows:

```
NDBID=database-type,database-id
```

3. Multiple Database Definition

Multiple database IDs of the same database type can be specified together, enclosed in parentheses:

```
NDBID=(database-type,database-id1,database-id2,...)
```

database-type

Possible Values	Explanation
DB2	Databases are accessed via Natural for DB2. This is the default value.

database-id

Possible Values
1-254

Special Requirements for Natural Tools for DB2

To be able to use the Natural Tools for DB2 (see *Using 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 DB2 catalog tables and they must be created as ALIAS, VIEW, or TABLE.

To help you create these tables, sample SQL code is provided in the member DEMSQL4 in the Natural system library SYSDB2. By default, it creates an ALIAS SYSSAG.xxx for the corresponding SYSIBM table.

For some catalog tables no indexes are defined. For performance reasons, 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
SYSFOREIGNKEYS
SYSINDEXPART
SYSKEYS
```

SYSSTMT
 SYSSYNONYMS
 SYSTABLEPART
 SYSVIEWS

The `CREATE TABLE` and `CREATE INDEX` statements required are included as comments in the sample SQL member `DEMSQL4`. In addition, `DEMSQLUP` includes sample SQL code to update the data in the copies of the catalog tables.

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

Note:

The sample SQL members can be executed with the `ISQL` part of `SYSDDB2`. `ISQL` enables you to read SQL members from the Natural system library `SYSDDB2`. 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 `SYSDDB2`.

LISTSQL and Explain Functions

These functions access `DB2 PLAN_TABLEs`.

 **To use these functions**

- Add an `LFILE` definition for logical file 101 to the `NATPARM` parameter module:

```
LFILE=(101,bdid,2)
```

where *bdid* is your `DBID` for `DB2`

A `PLAN_TABLE` must exist for your `SQLID`.

For the layout of the `PLAN_TABLE`, see the relevant `DB2` documentation by `IBM` of the `EXPLAIN` command.

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

`APPLNAME`
`PROGNAME`
`COLLID`
`QUERYNO`
`TIMESTAMP`
`DESC`
`QBLOCKNO`
`PLANNO`
`MIXOPSEQ`

Natural for DB2 Server Stub

A Natural for `DB2` server stub is an interface module needed to communicate between the `DB2` database system and the Natural server. The server stub module determines, sets up and invokes a Natural server environment for executing Natural stored procedures and Natural user-defined functions.

As mentioned in the Installation Procedure, there are two types of server stubs: the Natural for DB2 start server stub (STR) and the Natural for DB2 server stub (SRV). Both stubs are generated from the NDBSTUB macro.

- Natural for DB2 Start Server Stub
- Natural for DB2 Server Stub
- JCL Procedure
- Macro NDBSTUB

Natural for DB2 Start Server Stub

The Natural for DB2 start server stub is used for setting up the Natural server environments desired. The start server stub must be the main execution program in the Stored Procedure Address Space (SPAS). After the start server stub has established the Natural server environments, it passes control to the appropriate DB2 program (DSNX9WLM for WLM SPAS and DSNX9STP for DB2 SPAS). When SPAS terminates, the DB2 program returns control to the start server stub. The start server stub stops the Natural server environments and returns control to the operating system.

The Natural for DB2 start server stub reads the names and parameters of the Natural server to be started from the CMSRVIN dataset. CMSRVIN must be specified with DDNAME CMSRVIN.

The CMSRVIN dataset is a sequential file that contains all information required to start the desired Natural servers. For each server to be started, one START entry must be provided. The parameters used for the START entries are identical to the parameters that apply to the NDBSTUB macro. Enclose the contents of each START entry in brackets and delimit comments by the following signs: /* and */.

Example of START Entries:

```
START=( SERVER=WDB42SRV , NATURAL=NATBAT4R , CMPRMIN=CMPRMIN ,
        CMPRINT=CMPRINT , CMTRACE=CMTRACE , THREADSIZE=768 ,
        THREADNUMBER=2 , TRACE=ON )
START=( SERVER=WDB4SSRV , NATURAL=NATBAT4R , CMPRMIN=CMPRMIN ,
        CMPRINT=CMPRINT , CMTRACE=CMTRACE , THREADSIZE=768 ,
        THREADNUMBER=2 , TRACE=ON )
/* START=( SERVER=QE42SRV , NATURAL=NATBAT41 , CMPRMIN=QAPARM4 , */
/*          CMPRINT=CMPRINT , CMTRACE=CMTRACE , THREADSIZE=700 , */
/*          THREADNUMBER=2 , TRACE=OFF) */
```

If the start server dataset is missing or has not been assigned, the start server stub will start a Natural server environment with the parameters that derive from the parameters defined for the start server stub itself.

Natural for DB2 Server Stub

The Natural for DB2 server stub is the link between DB2 and Natural stored procedures or Natural user-defined functions (Natural UDFs). Specify the Natural for DB2 server stub as EXTERNAL_NAME in the SYSIBM.SYSROUTINES table row that refers to the Natural stored procedure or Natural UDF. The server stub is started by DB2/WLM when the Natural stored procedures or Natural UDFs are invoked. The Natural for DB2 server stub creates a Natural session in the Natural server environment and invokes the Natural subprogram comprising the Natural stored procedure or the Natural UDF.

A Natural session created for executing a Natural stored procedure terminates when the corresponding Natural subprogram ends and control returns to DB2 and to the calling client.

A Natural session created for executing a Natural UDF stays active for multiple function invocations if the PARALLEL attribute is set to D and the FINAL CALL attribute is set to Y. The session invoked for a Natural UDF function is terminated by the server stub if it detects a termination call.

JCL Procedure

The JCL procedure of the Stored Procedure Address Space (SPAS) must specify the Natural for DB2 start server stub as program in the EXEC statement.

The Natural for DB2 start server stub and the Natural for DB2 server stub must reside in a library contained in the steplib concatenation of the JCL procedure of the SPAS.

Example JCL:

```

//*****
//*      JCL FOR RUNNING THE WLM-ESTABLISHED STORED PROCEDURES
//*      ADDRESS SPACE
//*      RGN      -- MVS REGION SIZE FOR THE ADDRESS SPACE.
//*      DB2SSN   -- DB2 SUBSYSTEM NAME.
//*      NUMTCB   -- NUMBER OF TCBS USED TO
//*                PROCESS END USER REQUESTS.
//*      APPLENV  -- MVS WLM APPLICATION ENVIRONMENT
//*                SUPPORTED BY THIS JCL PROCEDURE.
//*
//*****
//DB27ENV2 PROC RGN=0K,APPLENV=DB27ENV2,DB2SSN=DB27,NUMTCB=8
//IEFPROC EXEC PGM=WDB42STR,REGION=&RGN,TIME=NOLIMIT, /* start server stub
/*IEFPROC EXEC PGM=DSNX9WLM,REGION=&RGN,TIME=NOLIMIT,
//      PARM=' &DB2SSN,&NUMTCB,&APPLENV'
//STEPLIB DD DISP=SHR,DSN=DSN710.RUNLIB.LOAD
//      DD DISP=SHR,DSN=CEE.SCEERUN
//      DD DISP=SHR,DSN=DSN710.SDSNLOAD
//      DD DISP=SHR,DSN=NATURAL.V2.TEST.NUCLEUS /* Library containing stubs and Natural nucleus
//CMPRMIN DD DISP=SHR,DSN=SAG.SYSF.SOURCE2(TDB31PRM) /* Dynamic Natural parameters.
//CMSRVIN DD DISP=SHR,DSN=SAG.SYSF.SOURCE2(CMSRVIN) /* Servers to be started.
//CEEDUMP DD SYSOUT=X
//SYSOUT DD SYSOUT=X /* Traces records of server stub
//RMTRACE DD SYSOUT=X
//CMPRINT DD SYSOUT=X
//SYSPRINT DD SYSOUT=X
//SYSERROR DD SYSOUT=X
//SYSUDUMP DD SYSOUT=X

```

Macro NDBSTUB

The NDBSTUB macro is used to generate the Natural for DB2 server stub and Natural for DB2 start server stub. You can parameterize NDBSTUB to create different stubs.

Below are the parameters available with NDBSTUB:

```

CMPRINT | CMPRMIN | CMTRACE | GTRACE | GTRCID | MAIN | MODE | NATURAL | SERVER |
THREADNUMBER | THREADSIZE | TRACE | WLM

```


CMPRINT - DDNAME of CMPRINT Dataset

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 Natural stored procedure is invoked.

Possible Values:

Value	Explanation
<i>ddname</i>	8 character DDNAME
CMPRINT	This is the default value.

CMPRMIN - DDNAME of CMPRMIN Dataset

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

Possible Values:

Value	Explanation
<i>ddname</i>	8 character DDNAME
CMPRMIN	This is the default value.

CMTRACE - DDNAME of CMTRACE Dataset

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

Possible Values:

Value	Explanation
<i>ddname</i>	8 character DDNAME
CMTRACE	This is the default value.

GTRACE - Natural for DB2 Server Stub to Execute GTRACE Calls

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

Possible Values:

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. This is the default value.

GTRCID - GTRACE ID to be Used

GTRCID specifies the event ID recorded with the trace data created by the Natural for DB2 server stub.

Possible Values:

Value	Explanation
<i>event-id</i>	Decimal number from 0 to 1023
203	This is the default value.

MAIN - No Longer Relevant and only Maintained for Compatibility Reasons

The value of MAIN is no longer evaluated. The Natural for DB2 server stubs check whether they are invoked as IBM LE (Language Environment) main program or as IBM LE subprograms and react accordingly.

Value	Explanation
YES	The generated server stub operates as IBM Language Environment main program. This is the default value.
NO	The generated server stub operates as IBM Language Environment sub program.

MODE - Operating Mode of Natural for DB2 Server Stub

MODE determines the operating mode of the Natural for DB2 server stub generated.

Value	Explanation
STR	The generated Natural for DB2 server stub operates as Natural for DB2 start server stub that sets up the Natural server environment.
SRV	The generated Natural for DB2 server stub operates as Natural for DB2 server stub that invokes the associated Natural stored procedure or Natural user defined function (UDF). This is the default value.

NATURAL - Name of Server Front-End or Natural Server

NATURAL denotes the name of the server front-end or Natural server load module which will be loaded by the Natural for DB2 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.

Value	Explanation
<i>name</i>	Any valid load module name
NATBATvr	This is the default value.

where *vr* is the current product version number.

SERVER - Server Name for Natural Server Environment

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 Natural server. These names must be unique within one address space.

Value	Explanation
<i>server-name</i>	Up to 5 characters
NDBvr	This is the default value.

where *vr* is the current product version number.

THREADNUMBER - No Longer Relevant and only Maintained for Compatibility Reasons

THREADNUMBER determines the number of Natural threads used by the Natural server. This number limits the number of Natural stored procedures and Natural UDFs concurrently active in the Natural server.

Note:

The value of THREADNUMBER is no longer evaluated. Instead, the Natural for DB2 start server stub uses the NUMTCB parameter of the SPAS JCL procedure as THREADNUMBER value. For further details, see the relevant DB2 literature by IBM.

Value	Explanation
<i>number</i>	Decimal number
10	This is the default value.

THREADSIZE - Size of Natural Threads for Natural Server

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

Value	Explanation
<i>threadsize</i>	Decimal number
768	This is the default value.

TRACE - Natural for DB2 Server Stub to Write Trace Rrecords

Determines whether the Natural for DB2 server stub generated writes trace records or not. The trace records are written to the dataset specified with DDNAME SYSOUT.

Value	Explanation
YES	Trace records are written.
NO	No trace records are written. This is the default value.

WLM - Natural for DB2 Start Server Stub Mode WLM/DB2 SPAS

WLM (Workload Manager) specifies where control is passed to after the Natural for DB2 start server stub has established the Natural server environments requested.

This parameter is only evaluated if the MODE parameter is set to MODE=STR. Specify WLM=YES if the Natural for DB2 start server stub runs in an address space that has been established by WLM.

Value	Explanation
YES	The start server stub generated links to DSNX9WLM, after setting up the Natural server environments.
NO	The start server stub generated links to DSNX9STP, after setting up the Natural server environments. This is default value.