

System File Structure

As described in section Accessing DL/I Data, certain information must be stored and maintained on the Natural FDIC system file in order to access DL/I data. This information describes the database structure, the segment data and the processing intent of an application. Four elements on the Natural FDIC system file contain this information. One of these elements, the Natural DDM, is common to all DBMS environments. The remaining three elements, however, are used only by Natural for DL/I; they are NDB control blocks, NSB control blocks and UDF control blocks. Therefore, the Natural FDIC system file used by Natural for DL/I contains three subfiles:

- The NDB Subfile
- The NSB Subfile
- The UDF Subfile

This section also provides information on:

- Natural for DL/I Objects
 - Displaying Keys of UDF Blocks
 - Displaying the Size of NDL Objects
 - Displaying NDL Objects
 - Control Blocks in Separate Buffer Pool
 - Control Blocks in Buffer Pool Blacklist
 - Natural for DL/I Objects and Natural DDMs
-

The NDB Subfile

The NDB subfile contains the NDBs. The NDB, or Natural DBD, control blocks contain most of the information present in the DL/I DBD, combined with additional data used by Natural, such as the file number (FNR) and database identification (DBID) of the segment, and short names for fields defined in the DBD. The NDB control blocks are created and stored on the Natural FDIC system file by the NATDBD procedure.

An NDB consists of the following fields:

| Field | Description |
|-------|---|
| ND | DBD name (8 characters) combined with sequence number (1 byte, "binary"). |
| NC | The first two bytes contain the number of NZ fields in the record times 20. The second two bytes contain the total number of NZ fields in the NDB multiplied by 20. |
| NZ | NDB data. |

The NSB Subfile

The NSB subfile contains the NSBs. The NSB, or Natural PSB, control blocks contain most of the information present in the DL/I PSB. These control blocks are created and stored on the Natural FDIC system file by the NATPSB procedure.

An NSB consists of the following fields:

| Field | Description |
|-------|---|
| NP | PSB name (8 characters) combined with sequence number (1 byte, "binary"). |
| NC | The first two bytes contain the number of NZ fields in the record times 20. The second two bytes contain the total number of NZ fields in the NSB multiplied by 20. |
| NZ | NSB data. |

The UDF Subfile

The UDF subfile contains the UDFs. The UDF, or User-Defined Field, control blocks contain information on segment fields which have been specified by the user, either through the online DL/I Services function of the SYSDDM utility, the NATUDF procedure, or by using Predict.

The fields are as follows:

| Field | Description |
|-------|--|
| NS | Database identification (1 byte, "binary"), file number (1 byte, "binary") and sequence number (1 byte, "binary"). The DBID and FNR are those of the segment being described by this record. |
| NC | The first two bytes contain the number of NZ fields in the record times 20. The second two bytes contain the total number of NZ fields in the UDF multiplied by 20. |
| NZ | Field description as specified by the user using Predict, the "EDIT segment layout" facility of SYSDDM or the procedure NATUDF. |
| NW | The long field name. |

Natural for DL/I Objects

Natural for DL/I objects are created during execution of the NATPSB procedure (NSB), during execution of the NATDBD procedure (NDB), or when assigning DBID/FNR to a segment type (UDF). Consequently, at least one UDF block for each segment type with an assigned DBID/FNR is always present on FDIC - whether User-Defined Fields (UDF fields) have been defined by the user or not.

When displaying type definitions in SYSDDM, the NDB and its related UDF are combined automatically. The only way to display an UDF separately (for debugging purposes) is by using NDLBLOCK.

Displaying Keys of UDF Blocks

The utility program NDULIST, cataloged in the library SYSDDM, is provided for listing the keys of all UDF blocks and for checking for duplicates.

For each duplicate found the following warning is issued:

More than one record with same DBID/FNR.

Displaying the Size of NDL Objects

The following utility programs, cataloged in library SYSDDM, are provided for displaying the sizes of the various NDL objects:

- NDLSIZED displays the sizes of all NDBs stored on FDIC.
- NDLSIZEP displays the sizes of all NSBs stored on FDIC.
- NDLSIZEU displays the sizes of all UDFs stored on FDIC.

Displaying NDL Objects

The utility program NDLBLOCK, cataloged in library SYSDDM, is provided for displaying the NDBs, NSBs and UDFs stored on FDIC. The utility displays the objects in hexadecimal format.

Control Blocks in Separate Buffer Pool

The Natural for DL/I control blocks NDB, NSB and UDF are read from FDIC and loaded into a buffer pool - resident or not, depending on the NDLPARM parameters RESINDB, RESINSB, and RESIUDF. This allows a given object to be shared by several users.

By means of the NTBPI macro (as described in Parameter Modules in the Natural Parameter Reference documentation) it is possible to have a buffer pool for NDB, NSB and UDF control blocks which is different from the buffer pool for Natural programs, thus allowing for better isolation between the different Natural objects.

If a separate buffer pool is allocated, Natural for DL/I locates its control blocks in this buffer pool. Otherwise, they are located in the Natural buffer pool.

The "Individual Object Statistics" function of the SYSBPM utility displays the NDB, NSB and UDF control blocks kept in the buffer pool as follows:

| | Library | DBID | FNR |
|-----|-----------------|-------------|------------|
| NDB | SYSDLIND | 255 | 253 |
| NSB | SYSDLINS | 255 | 253 |
| UDF | <i>Ummmmnnn</i> | 255 | 253 |

Note:

The library names of NDB and NSB are fixed internal names and are not related to any Natural library.

The DBID/FNR values are fixed internal values and are not related to any Natural system file.

"mmm" is the DBID of the corresponding segment, "nnn" is the FNR of the corresponding segment.

The "Display Object Hexadecimally" function of the SYSBPM utility also allows you to display Natural for DL/I objects. This function might be useful when in doubt if the expected object has been read from FDIC, or if the object has been read from the expected FDIC (test/production).

Control Blocks in Buffer Pool Blacklist

The Natural for DL/I control blocks NDB, NSB and UDF can be added to the buffer pool blacklist.

This is done by the "Blacklist Maintenance" function of the SYSBPM utility.

As "Library" you enter "SYSDLIND" for NDBs, "SYSDLINP" for NSBs, and "SYSDLINS" for UDFs.

As "Object" you enter the NDB name for NDBs, the NSB name for NSBs, and *Ummmmnnn* for UDFs where *mmm/nnn* are the DBID/FNR of the corresponding segment.

This feature allows you to modify NDBs, NSBs or UDFs without causing unpredictable results for active users.

If an attempt is made to load a locked object into the buffer pool, Natural for DL/I will issue error message NAT3935.

Natural for DL/I Objects and Natural DDMs

When referencing a DDM in a Natural program, Natural translates the DDM name into the corresponding DBID/FNR pair. If this DBID identifies the DDM as a DL/I DDM (by means of the NTDB macro), the Adabas control block is passed to Natural for DL/I for further processing.

Natural for DL/I takes DBID from the control block and tries to locate an UDF with this DBID/FNR in the buffer pool. If it is not found there, it is read from FDIC and loaded into the buffer pool.

The UDF contains the name of the related NDB in its header. Using this name, Natural for DL/I tries to locate the NDB in the buffer pool. If it is not found there, it is read from FDIC and loaded into the buffer pool.

The segment description including all DL/I fields is part of the NDB.

From this it is clear that:

- the NDB/UDF is required during runtime,
- the relation between the Natural program and the related NDB is established by means of DBID/FNR only.

This implies that the DBA has to ensure that DDMs and NDBs are always kept in synchronization. For example, it is not sufficient to transfer only the Natural programs from test to production.