

DL/I Services

When you invoke "DL/I Services" from the SYSDDM main menu, the **DL/I Services Main Menu** is displayed which offers you the following functions:

-

NDB Maintenance

An NDB is a DL/I DBD (database description) which is defined to Natural.

-

NSB Maintenance

An NSB is a DL/I PSB (program specification block) which is defined to Natural.

NDB Maintenance

This section covers the following topics:

- Menu and Functions
- Select an NDB from a List
- Select an NDB Segment from a List
- Edit an NDB Segment Description
- Generate DDM from Segment Description

Menu and Functions

When you select **NDB Maintenance** on the **DL/I Services Main Menu**, the **NDB Maintenance** menu is displayed:

```
14:37:12                **** DL/I Services ****                2006-05-25
                        - NDB Maintenance -

                        Code Functions
                        -----
                        S  Select an NDB from a List
                        P  Purge an NDB
                        L  Select an NDB Segment from a List
                        E  Edit an NDB Segment Description
                        G  Generate DDM from Segment Description
                        ?  Help
                        .  Back
                        M  End
                        -----

Enter Code: ?
NDB Name:
Segment Name:

ENTER  PF1  PF2  PF3  PF4  PF5  PF6  PF7  PF8  PF9  PF10  PF11  PF12
      Help      Back                                End
```

The individual NDB maintenance functions are listed below:

Function	Explanation
Select an NDB from a List	<p>List the NDBs which are defined on the Natural system file. You can then select NDBs from this list by entering the following function codes:</p> <p>P to purge an NDB, L to list the segments of an NDB.</p> <p>For details, see <i>Select an NDB from a List</i>.</p>
Purge an NDB	<p>Purge an NDB and its related segment descriptions from the Natural system file. The name of the NDB to be purged must be specified.</p> <p>Before this function is executed you are prompted to confirm the purge request.</p> <p>For details, see <i>Select an NDB from a List</i>.</p>
Select an NDB Segment from a List	<p>List the segments of the specified NDB. You can then select segments from this list for further processing.</p> <p>For details, see <i>Select an NDB from a List</i>.</p>
Edit an NDB Segment Description	<p>Edit a segment description. The segment name and its corresponding NDB name are required when invoking this function. A database ID (DBID) and file number (FNR) must have been assigned to the segment description (function code A on the Segment List display) before it can be edited.</p> <p>For details, see <i>Edit an NDB Segment Description</i>.</p>
Generate DDM from Segment Description	<p>Generate a DDM from a segment description. The DDM definition is a Natural DDM of the segment. Prior to execution of this function, a DBID and FNR must have been assigned to the segment (function code A on the Segment List display).</p> <p>For details, see <i>Generate DDM from Segment Description</i>.</p>

Select an NDB from a List

When you select an NDB from a list, a list containing all NDBs defined on the Natural system file is displayed. In addition to the NDB name the following is displayed:

L/P	Indicates if ACCESS=LOGICAL or not.
length	Length of the NDB.
NoSGMS	Number of the segment types in the NDB.
ACCESS	The access specification taken from the DBD.

From the list, you can select NDBs for further processing by entering the following function codes in the **Func** column next to the NDB names:

Code	Function
P	<p>Purge NDB</p> <p>This function is identical to the Purge NDB function available on the NDB Maintenance menu. It deletes an NDB and its related segment descriptions from the Natural system file. Before the function is executed you are prompted to confirm the purge request.</p>
L	<p>List NDB Segments</p> <p>This function is identical to the "Select NDB Segment from a List" function available on the NDB Maintenance menu. It lists the segments of the selected NDB.</p> <p>For details, see <i>Select an NDB Segment from a List</i>.</p>

Select an NDB Segment from a List

When you select an NDB segment from a list, a list containing all segments of the specified NDB is displayed. If you do not know the NDB name, use the **Select an NDB from a List** function.

```

10:50:48                **** DL/I SERVICES ****                2006-05-25
                        - Segment List -

DBD Name = ED00DBD
  Func  Level  Segment      DBID  FNR    Seg-Lgh      UDF-Lgh      Response
-----
_       1     COURSE       246   _10    75-80        100
_       2     PREREQ       246   _11    36-36        40
_       2     OFFERING     246   _12    41-41        40
_       3     TEACHER      246   _13    24-24        60
_       3     STUDENT      246   _14    40-40        40
_
_
_
_
_
----- Bottom -----
Code .. _ ( ? Help  . Back  M End )
Func = E Edit Segment Description  A Assign DBID and FNR
      F Free DBID and FNR          ' ' Change DBID and FNR
      G Generate DDM               N Take New Copy of UDF

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Exec Help          Exit                               Canc

```

Next to each segment you can enter one of the function codes listed below. You can mark several segments at the same time with a function code. If you do not enter any code, the list is scrolled forward until the bottom of the list is reached.

You can enter one of the following codes next to a segment on the segment list to perform one of the following functions:

Code	Function
A	<p>Assign DBID/FNR</p> <p>Assign a DBID and a FNR to the selected segment.</p> <p>The DBID is a number in the range from 1 to 254. It must be contained in the database ID list (NTDB macro) of the Natural parameter module NATPARM. All Natural DDMs which refer to a DL/I segment must have a DBID belonging to this range. If a DBID has not been entered, a default value is assigned, which is the entry with the lowest value in the DBID list specified in the Natural parameter module. For a given NDB, all segments should be assigned the same DBID. Otherwise, Natural assumes different databases and generates an OPEN command (which, however, is ignored by the DL/I call handler).</p> <p>The FNR is a number in the range from 1 to 254. The FNR must be specified; no default value is assumed by Natural. The segments of a logical NDB must have file numbers different from those assigned to the segments of the physical NDB.</p> <p>The DBID/FNR combination must be unique on the Natural system file. It is used by Natural to uniquely determine the NDB and the segment within the NDB.</p>
E	<p>Edit Segment Description</p> <p>Edit the description of a segment within a given NDB. The function is the same as the Edit an NDB Segment Description function which you can invoke from the NDB Maintenance menu. Before you can edit a segment description, a DBID and FNR must have been assigned to the segment (see above).</p> <p>For details, see <i>Edit an NDB Segment Description</i>.</p>
F	<p>Free DBID/FNR</p> <p>Release the DBID and FNR which have previously been assigned to the segment. Once a DBID and FNR have been released, they are available for assignment to another segment.</p>
G	<p>Generate DDM</p> <p>Generate a DDM definition from a segment description. The function is the same as the Generate DDM from Segment Description function which you can invoke from the NDB Maintenance menu.</p> <p>Before you can generate a DDM from a segment description, a DBID and FNR must have been assigned to the segment (see above).</p> <p>The generated DDM is a Natural view of the segment. Once it has been generated, the DDM can be modified and cataloged.</p> <p>For details, see <i>Generate DDM from Segment Description</i>.</p>
N	<p>Take New Copy of UDF</p> <p>Refresh the user-defined fields (UDFs) of a segment when the UDFs of the source segment have been changed. This applies to UDFs of segments belonging to a logical NDB and to UDFs of logical virtual children. Before you can execute this function, a DBID and FNR must have been assigned to the segment (see above).</p>

Code	Function
<i>blank</i>	<p>Change DBID/FNR</p> <p>Change a previously assigned DBID and/or FNR.</p> <p>For changing a DBID or FNR, the same rules concerning DBID and FNR specification apply as for assigning a DBID/FNR (see above).</p>

Edit an NDB Segment Description

Additional segment fields, so-called user-defined fields (UDFs), can be defined.

This function is invoked either by entering function code E, an NDB name and a segment name on the **NDB Maintenance** menu, or by selecting the segment from the **Segment List** (by marking it with function code E). A DBID and a FNR must have been assigned to a segment description (function code A on the **Segment List** display) before it can be edited.

EDIT command:		DBD	ED00DBD	SEGMENT	STUDENT	SEGLGH	40-40
ALL	LEV	SN	FIELD NAME	START	DLI	MAXOCC	FOR LGH V
	1	PM	EMPNO	00001	SQU	A	6
	1	PN	NAME	00007	SRC	A	33
	1	PO	GRADE	00040	SRC	A	1
	1	AA	BIRTHDATE	00025		A	
	2	AB	DATE-DD			N	2
	2	AC	DATE-MM			N	2
	2	AD	DATE-YY			N	2
	1	AE	BIRTHPLACE			A	10
	1	AF	STUDENT-NAME	PN		A	18

The following information is displayed on the status line at the top of the screen:

DBD	Name of the DBD which contains the edited segment.
SEGMENT	Name of the edited segment.
SEGLGH	Minimum and maximum length of the edited segment, separated by a hyphen.

DL/I fields and user-defined fields are displayed as shown above. You can add, delete or modify UDFs. DL/I fields, however, can neither be added nor deleted. If the specification `TYPE=P` is included in the `FIELD` statement of the DL/I DBD, the format of the field can be changed from P (decimal packed unsigned) to S (decimal packed signed) on the edit segment description screen. `FOR` (format) is the only attribute of a DL/I field you can modify. In particular, it is not possible to change the name of a DL/I field, because it is used by Natural to build the segment search arguments (SSA). If the name of a DL/I field is to be changed, the field can be redefined as an UDF.

Edit commands are available to copy or delete single lines or to insert a group of empty lines. In addition, commands for scrolling forward or backward are provided. For details you can enter a question mark in the "command" field to display the corresponding help information.

After modification of segment field attributes you can save the description by entering `SAVE` in the command field.

The following field definition attributes are displayed and can be modified for user-defined fields:

Attribute	Description
LEV	Level number used to define a group of fields.
SN	Short name of the field as used internally by Natural.
FIELD NAME	Name of the field as used in the application programs.
START	Start position of the field in the segment.
DLI	Type of the DL/I field, as follows: SIX secondary index field SQU sequence field (unique) SQM sequence field (multiple) SRC search field
MAXOCC	Maximum number of occurrences of a multiple field or periodic group.
FOR	Format of the field.
LGH	Length of the field.
V	Variable field length indicator.

Each user-defined field can be defined as follows:

Field Type	Description
Elementary Field	A field that contains only one value in a single segment. Example: Personnel number
Multiple Field	A field that can contain more than one value in a single segment. Reference to a particular value of a multiple field can be made by appending a one to three-digit subscript (value 1 - 191) to the field name. Example: Languages - English, German, Italian
Group	A series of one or more adjacent fields that can be referenced with a single name (the group name). You can also refer to a single field of a group by specifying its name. Example: 01 Address Group field 02 City Elem. field 02 Street " " 02 Number " "
Periodic Group	A group which is repeated in multiple adjacent occurrences in a single segment. For a periodic group it is possible to refer to a range of occurrences (or a field within a periodic group) by specifying the first and the last occurrence number to be referenced (connected by a hyphen (-)) after the name and in ascending order. Multiple-value fields or periodic groups are not allowed within a periodic group. Example: Several addresses

Since DL/I fields cannot be modified as described above (with the exception of `FORMAT`), they cannot be directly defined as a group. To define a DL/I field as a group, it is necessary to redefine it as a user-defined field which then can be redefined as a group. In a DDM, these user-defined fields must not be specified as descriptor fields. When a DDM is generated, the UDFs are marked as non-descriptor fields.

Example - Redefinition of a DL/I Sequence Field as a Group:

The description of the segment `STUDENT` within the DBD named `ED00DBD` is used as shown in the Segment List screen above:

LEV	SN	FIELD NAME	START	DLI	NOCC	FOR	LGH	V
1	PM	EMPNO	00001	SQU		A	6	
.								
.								
.								

If the DL/I sequence field `PM` is to be "structured", it must be redefined as a user-defined field (`AAAAA` in the figure below). This UDF can then be structured as required.

LEV	SN	FIELD NAME	START	DLI	NOCC	FOR	LGH	V
1	PM	EMPNO	00001	SQU		A	6	
.								
1	AA	AAAAA	PM					
2	AB	BBBBB				A	3	
2	AC	CCCCC				A	3	
.								
.								
.								

The group field AAAAA has no format/length (FOR/LGH) specified. The length of a group is set equal to the sum of all fields belonging to the group.

UDF Parameters

For each user-defined field on the above screen, parameters can be specified as listed and described in the following table. The total length of all DL/I fields and user-defined fields must not exceed the segment length.

When attributes of a UDF are modified and an old copy of this UDF is contained in the shared UDF buffer pool, the old copy is marked "invalid". If the UDF is referred to again by a Natural program, the modified UDF is read from the Natural system file. Therefore, it is not necessary to restart the Natural session if a UDF has been modified. However, this applies only to physical UDFs; that is, to UDFs of a physical NDB. If a physical UDF is modified and a logical NDB refers to the appropriate segment type, the logical UDF is not marked "invalid" in the buffer pool. To invalidate a logical UDF it is necessary to restart the TP monitor or to execute function N (**Take New Copy of UDF**) of the **Segment List** screen on the appropriate segments in the logical NDB.

Field	Description
LEV (level number)	A one-byte value used to define a group. A field is a group only if the subsequent field has a higher level number. The field immediately after the last group element must have a lower level number. A group can be defined within another group. The level number of the first user-defined field must be 1.
SN (short name)	The name used internally by Natural to identify the field. It must be two bytes in length, the first character must be alphabetic in the range from A to G (E is not permitted). The second character can be alphanumeric (that is, up to 216 UDF fields can be defined). If the segment is a logical child, the first character must be alphabetic in the range from H to M. Short names must be unique among a segment type.
FIELD NAME	External field name, up to 19 bytes long.

Field	Description
START	<p>The start position of the field in the segment. The position can be specified as absolute by giving a three-digit number or it can be specified as relative, by giving the short name of a previously defined field which is being redefined.</p> <p>It is important to specify the start position for the first user-defined field; otherwise, a default of 1 is used, which may cause overlapping with previous DL/I fields. The default for all other user-defined fields is the position immediately after the previous field.</p> <p>The redefinition of fields is possible only for fields which have the same level number. When the level is higher than 1 (that is, for a field inside a group), only the last field can be redefined with the same level number. An absolute position must not be specified for a field within a group.</p>
MAXOCC	<p>The maximum number of occurrences of a multiple -value field or periodic group in a segment.</p>
FOR (format)	<p>Standard field formats are:</p> <ul style="list-style-type: none"> A Alphanumeric B Binary F Fixed Point P Packed decimal unsigned; that is, the zone halfbyte of the last byte is X ' F ' . S Packed decimal signed; that is, the zone halfbyte of the last byte is X ' C ' (positive) or X ' B ' (negative). N Unpacked
LGH (length)	<p>Field length is a three-digit number; it must not exceed the maximum length permitted. These are as follows:</p> <ul style="list-style-type: none"> 253 bytes for alphanumeric fields (A), 126 bytes for binary fields (B), 4 bytes for fixed point (F), 14 bytes for packed decimal unsigned (P), 14 bytes for packed decimal signed (S), 27 bytes for unpacked decimal (N) <p>In addition, the length specified must not exceed the segment length. Length must not be specified for a group. The length of packed fields is the field length in bytes.</p>

Field	Description
V (variable)	<p>Depending on its value, V or blank, this parameter indicates whether a field has a variable length. Fields can be specified as variable only if the segment is a segment of variable length.</p> <p>Only one field can be defined as variable within a given segment description.</p> <p>An elementary field can be specified as variable in length only if it is the last field in the segment. A multiple field or a periodic group can be specified as variable in length regardless of its position in the segment.</p> <p>When applied to a multiple field or a periodic group, a setting of VARIABLE means that the number of occurrences is not known at definition time; therefore, MAXOCC should be specified using the maximum expected value.</p>

Generate DDM from Segment Description

This function is invoked either by using the G function code of the **NDB Maintenance** menu - then an NDB name and a segment name must be specified -, or by selecting the segment from the **Segment List**, by marking it with function code G.

A DBID and a FNR must have been assigned to a segment description (function code A on the **Segment List** display) before a DDM can be generated.

The DDM is generated from a segment description and represents a Natural view of the segment. It must be generated and cataloged before the corresponding segment can be referenced by a Natural program. After generation, default options for field headers or edit masks (decimal positions) can be modified in the DDM. See *Catalog DDM* and *Edit DDM* in the Natural *Utilities* documentation for corresponding information.

It should be noted, however, that default options for field headers or edit masks (decimal positions) are stored with the DDM and not with the NDB or UDF. The data in the NDB or UDF reflects what is allowed by the DL/I FIELD macro in which the length can be specified only in bytes (decimals are not allowed). Consequently, when regenerating the DDM, prior modifications in the DDM must be applied again by the user.

In DL/I a program must be able to reference search fields, sequence fields and secondary index fields of ancestor segments in order to build a certain search criterion; therefore, DDMs for DL/I segments can also include fields which are not part of the actual physical segment.

To satisfy the requirements for DL/I processing, a DDM must contain all the fields which can be referenced. Therefore, the generated DDM can contain the following fields:

- DL/I sequence fields, search fields and secondary index fields of the current (physical) segment. These fields have been defined in the DBDGEN source for this segment. When the DDM is generated, information on these fields is obtained from the NDB control block for this segment. DL/I sequence fields and secondary index fields are marked as descriptor (D), search fields are marked as non-descriptor (N). All of these fields can be used to qualify search requests.

- DL/I sequence fields and secondary index fields of all the ancestor segments. These fields have been defined in the DBDGEN source for the ancestor segments. When the DDM is generated, information on these fields is obtained from the NDB control blocks for the ancestor segments. These fields are marked as descriptor (D). They can be used to qualify search requests.
- DL/I search fields of all the ancestor segments. These fields have also been defined in the DBDGEN source for the ancestor segments. When the DDM is generated, information on these fields is also obtained from the NDB control blocks for the ancestor segments. However, these fields are marked as superdescriptor (S). They can be used to qualify search requests.
- Fields of the current segment defined by the user (UDFs). When the DDM is generated, information on these fields is obtained from the UDF control blocks. These fields cannot be used to qualify search requests.

Fields of format S in the segment description (see *UDF Parameters*) generate format P in the DDM.

The following tables summarize how the various types of fields can be processed using Natural I/O statements. They illustrate which fields can be used to qualify search requests, and which fields can be used with the Natural statements DISPLAY, UPDATE or STORE. In addition, the tables indicate whether the field in the generated DDM is marked as descriptor, superdescriptor or non-descriptor.

Current Segment

Type of field	FIND/READ	DISPLAY	UPDATE	STORE	Marked
DL/I sequence	yes	yes	no	yes	D
DL/I search	yes	yes	yes	yes	D
DL/I SIX	yes	yes	no	no	D
UDF	no	yes	yes	yes	blank

Ancestor Segment

Type of field	FIND/READ	DISPLAY	UPDATE	STORE	Marked
DL/I sequence	yes	yes	no	yes	D
DL/I search	yes	no	no	no	S
DL/I SIX	yes	yes	no	no	D
UDF	no	no	no	no	blank

Notes:

1. Using the Natural statement DISPLAY, the DL/I SIX fields can be displayed only if a PCB is used with this SIX specified in the PROCSEQ parameter. If not, an error message is returned by Natural at runtime.
2. The DL/I SIX field name cannot be used in an UPDATE or STORE statement. SIX fields, however, can be updated/stored by referring to the source fields which comprise the SIX.
3. The READ statement returns records in ascending sequence. The possible sequences for DL/I segments are root sequence or the sequence of any secondary index.

As mentioned above, the generated DDM contains all fields of the current segment and all DL/I fields of the ancestor segment(s), marked either as D or S. The UDFs of the ancestor segments are not included in the generated DDM because a DDM refers only to one segment.

The generated external name of the DDM is equal to the segment name prefixed by the DBD name.

Example:

Name of DBD:	ED00DBD
Name of segment:	STUDENT
Name of generated DDM:	ED00DBD-STUDENT

The generated external name of DL/I fields is equal to the name specified in the DL/I FIELD macro during the DL/I DBDGEN procedure.

The generated external name of DL/I fields of ancestor segments is equal to the field name suffixed by the segment name.

Example:

Name of DL/I field:	LOCATION
Name of ancestor segment:	OFFERING
Name of generated field:	LOCATION-OFFERING

The generated external name of the UDFs is equal to the name specified by the user at definition time.

NSB Maintenance

When you select **NSB Maintenance** on the **DL/I Services Main Menu**, the **NSB Maintenance** menu is displayed.

From this menu, you can select the following NSB maintenance functions:

Function	Explanation
Select an NSB from a List	List the DL/I PSBs defined on the Natural system file. You can select NSBs from this list by entering the function code P to purge an NSB, or L to list all PCBs and SENSEGs of an NSB.
Purge an NSB	Delete an NSB and its related PCB descriptions from the Natural system file. The name of the NSB to be deleted must be specified. Before this function is executed, you are prompted to confirm the deletion.
List PCBs and SENSEGs of an NSB	For any NSB specified, this function lists the PCBs and their sensitive segments. If an indexed database exists, its name is displayed under the header "PROCSEQ".

Select an NSB from a List

```

10:44:50                **** DL/I SERVICES ****                2006-05-25
                        - NSB List -

      Func   NSB Name  CMPAT  Length  NoPCBs  Response
      -----  -----  -----  -----  -----
      _      DFSIVP6   YES    140    3
      _      PBNDL01   NO     160    3
      _      PBNDL02   YES    160    1
      _      PBNDL03   YES    160    3
      _      PBNDL04   YES    160    1
      _      PBNDL05   NO     80     1
      _      PBNDL97   YES    160    3
      _      PBNDL98   YES    200    5
      _      PBNDL99   NO     200    5
      _      PBPQA01   YES    60     5
      _      PBSUP06   NO     440    5
      -----  -----  -----  -----  -----
                        - More -
Code .. _ ( ? Help, . Back, M End )

Func .. P (Purge NSB) L (List PCBs and SENSEGs)

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Exec  Help           Exit                               Canc

```

List PCBs and SENSECs of an NSB

```

10:46:57                **** DL/I SERVICES ****                2006-05-25
                        - PCB List -
NSB Name: PBNDL01 (CMPAT=NO ,Length=00160)

Number of PCB's  NDB Name      Level  SENSEG      PROCSEQ
-----
3                ED00DBD
                        1      COURSE
                        2      PREREQ
                        2      OFFERING
                        3      TEACHER
                        3      STUDENT

                        ----- Bottom -----

Code .. _ ( ? Help . Back M End )

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Exec  Help          Exit                               Canc

```

```

10:49:10                **** DL/I SERVICES ****                2006-05-25
                        - NDB List -

Func      NDB Name  L/P  Length  NoSGMs  Access  Response
-----
_         CCCBTD00  P    460     6
_         DNDL01  P    540     5
_         DNDL02  P    620    10
_         DNDL03  L    820    10
_         DNDL04  P     60     1   GSAM
_         DPQA04  P    480     5
_         DSUP02  L   1720    15
_         DSUP05  P    380     5
_         DSUP09  P    340     2
_         DSUP10  L    880    10
_         DUSA01  P    320     5   HDAM

                        ----- More -----

Code .. _ ( ? Help, . Back, M End )

Func: P (Purge NDB) L (List NDB Segments)

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Exec  Help          Exit                               Canc

```