Tables - Configuration Dialog Box

Database tables - configuration dialog box

Editing of all objects in the process D2000 CNF is being performed in the configuration dialog box, a specific part of which is common for all editable objects and another part depends on the type of edited object.

Configuration dialog box of objects of *Database table* type consists of several parts (tabs), which contain similar parameters.

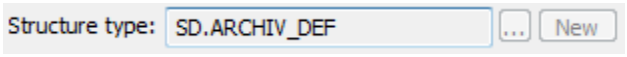
General properties

Description

A text string describing the database table. Maximum: 128 characters.
Possibility to use the Dictionary (to open press **CTRL+L**).

Table

Structure type



An object of Structure definition type, which defines the structure type. If the database table is *being used*, then it is not possible to change the structure type.

The **New** button enables to create a new structure definition according to the table definition in the database. The name of structure definition is derived from the name of *DB Table* object in D2000 System.
(The relevant system process D2000 DBManager must be running when creating the new definition.)

States when creating the new structure definition

- *Definition is not in the system* - the system requires the definition of database table and creates the new structure definition. After saving the definition, the system ensures the connection of definition to the *DB Table*.
- *Definition is in the system* - after confirming, the system ensures connection of definition to the *DB Table*.

Conversion table

D2000 types	ODBC	OCI
BOOL	SQL_BIT	
INT	SQL_INTEGER, SQL_NUMERIC, SQL_BIGINT, SQL_SMALLINT, SQL_TINYINT	SQL_INTEGER, SQL_NUMERIC, SQL_BIGINT, SQL_SMALLINT, SQL_TINYINT TYPE_NUMBER, TYPE_INTEGER, TYPE_UNSIGNED8, TYPE_UNSIGNED16, TYPE_UNSIGNED32, TYPE_SIGNED8, TYPE_SIGNED16, TYPE_SIGNED32, TYPE_SMALLINT
REAL	SQL_DECIMAL, SQL_FLOAT, SQL_REAL, SQL_DOUBLE, SQL_NUMERIC	TYPE_FLOAT, TYPE_DECIMAL, TYPE_REAL, TYPE_DOUBLE, TYPE_NUMBER
TEXT	SQL_CHAR, SQL_VARCHAR, SQL_WCHAR, SQL_WVARCHAR	TYPE_VARCHAR, TYPE_VARCHAR2, TYPE_CHAR
TIME	SQL_DATE, SQL_TIME, SQL_TIMESTAMP	TYPE_DATE, TYPE_TIME, TYPE_TIME_TZ, TYPE_TIMESTAMP, TYPE_TIMESTAMP_TZ, TYPE_INTERVAL_YM, TYPE_INTERVAL_DS, TYPE_TIMESTAMP_LTZ

Mapping columns between the table in the database and the columns in the structure definition is carried out by name. When generating SQL D2000 commands, D2000 DBManager captures names into quotation marks by default. In some cases, this is unwanted activity, so the D2000 DBManager process have the /NQ parameter.

Access

Selection of access rights to the database.

- None - no access to the database
- Read only - the database cannot be modified (only read)
- Modify - the database can be read and modified

Table

Name of the table in the database. This name can be either simple (e.g. *table1*) or compound. Compound name of table consists of dot separated user name and simple name of table (e.g. *user1.table1*). Compound names are supported by e.g. MsSql, Oracle and Sybase, simple names are required by e. g. Microsoft Access databases and MySql. There are following rules for working with table:

- If the name is simple, it is automatically extended with the user name (the parameter User) defined for the parent object of Database type. If the user name is not configured, the simple name of table is used.
- If the name is compound, it is directly used.
- If the name is in the form of *.table1* or *"".table1*, the simple name *table1* is used and it is not extended with the user name (the parameter User) defined for the parent object of Database type.
  Note: this rule is not valid for dbmanager_ora.exe which always uses the compound name of table.

Note: The name of table (including the name of user separated by comma) can be up to 64 characters. The length of table name is given by a particular database.

Index

Column (columns), which is (are) regarded as a key item. Key item is such an item, which uniquely defines a row in the database. The list of possible key items is equal to the names of columns according to *Structure definition*. The parameter is optional.

Optional

Column (columns) that is (are) considered to be optional. The optional column need not exist in a database. The list of possible optional columns is equal to the columns in *Structure definition*. To verify the existence of required columns in the database table (i.e. all that were not marked as optional), use Test button for the object of *Table* type or Test table for the object of *Database* type.

Not Null

Column (columns) that is (are) considered to be NOT NULL.
The value of NOT NULL column must be defined before inserting or modification in the database table (the operation to insert or modify the record tables). The list of possible NOT NULL columns is equal to the columns defined in *Structure definition*. If there are undefined values in these columns before inserting or modifying of tables, the action is terminated with error. All the values that do not comply with the NOT NULL condition, are listed, however, at most 10 for 1 column (it is in contrast with the DB engine - it returns only first conflicting column in the first conflicting row).

Example of error message, which is displayed in DBManager:

*%D2DBM-E-*** Error in con 1:*
%D2DBM-E-con 1: DBS_INSERT : Column "column1" [row # 7], "column3" [row # 3 7 8], "column5" [row # 1 2 3 4 5 6 7 8 9 10 ...] in table "dba"." test_js_column_multi" cannot be NULL!

Data category

Allows assigment of data category from business point of view to table column using Data category object type. By assigning data category to column a reference to assigned object comes into existence. It is then easily possible to find all usages of given data category. By assigning data category, attribute *Export monitored* is automatically selected.

Data purpose

Assignment of Data purpose object into column activates anonymization process on given column. Anonymization is automatic process which modifies values in anonymized column to which processing time as given in assigned data purpose has expired. New value will be set using value given in replacement attribute. Anonymization is run periodically, by default every hour and is performaed by D2000 DBManager process. For each column, which has data purpose assigned, those rows will be anonymized which value in time column increased by processing time is smaller than current time and is not between anonymized time intervals. By assigning data category, attribute *Export monitored* is automatically selected.

Time column

Attribute which marks absolute time type column of configured table, from which processing time of data in given column is measured. Attribute is mandatory if data purpose is defined.

Replacement

Attribute which defines value that will be set into column during process of anonymization. If value is not defined, empty (null) value is set which can be interpreted as deletion of value. For textual values it is possible to use combination of predefined text and mask of date and time which will be replaced by actual values of date and time at time of anonymization. The mask is written between brace brackets and uses the same characters as ESL function % TimeToStr. It is possible to use time mask multiple times in text, e.g.: "Anonymized on {dd.mm.yyyy} at {hh:mi:ss}.". For values of absolute time type only mask format {hh:mi:ss dd-mm-yyyy} can be used or actual date and time value using the same format or empty (null) value.

Test

The button allows testing the database table connection functionality. Before running the test, it is necessary to click the button **Save**, if you performed any changes in the parameters **Structure type**, **Access** or **Table**.
Testing requires that process D2000 DBManager must be running.

When testing a database table, process D2000 DBManager reloads the table definition from the SQL database using the ODBC function **SQLColumns**. The feature can be used for working in on-line system, when adding a column/columns into a table in the SQL database (the column/columns already exist in the Structure definition in D2000 system) and it is necessary, that process D2000 DBManager reloads the table definition in the SQL database for working with the added column/columns. Clicking the button **Test** reloads the table definition from the SQL database and the column/columns added is to be also taken into account. Tables that have been already opened (DB_CONNECT, PG_CONNECT) will use "original" columns (columns that exist at the time of opening the tables), tables opened later can use the added columns.
An alternative to this method is restarting process D2000 DBManager. If you add a column into a Structure definition in the D2000 system, process D2000 DBManager automatically reloads the table definition from the SQL database.
If the test is successful (and the table is found), process D2000 DBManager notifies the process D2000 CNF about the successful result. In case, that some columns of the structure definition are not included in the table in SQL database, process D2000 DBManager shows a warning containing the list of columns that were not found.
If a column is table is defined as a text one in SQL database, but its type in respective structure definition is different, the warning is shown, as well. That behaviour has been implemented for Oracle database (e.g. a column is of *Int* type defined in D2000 system and the same column is of *VARCHAR* type in Oracle database then paging respective table in D2000 may not be able to show all pages correctly).
If some columns, that are not defined as optional in the configuration, are not in the table, they are listed in the form of warning.

Note: By using *dbmanager.exe* (ODBC version), the test is not to be successful if the first row that is read from table in SQL database contains a text column that cannot be converted to non-text type. The problem is in ODBC driver (current version of Oracle ODBC 9.02.00.65). If the first row is correct (or the table contains no row), error message is shown correctly.
By using *dbmanager_ora.exe* (OCI version), this problem does not occur because D2000 DBManager unlike Oracle ODBC driver correctly handles error states generated by the OCI layer.

SQL definition

The button **Copy to clipboard**, depending on the access type to database, enables to copy the proper Oracle SQL definition to a clipboard. If there is set read-only access, VIEW definition, which contains the columns from a structure definition, will be stored. Otherwise, SQL command will be stored to create a table of the appropriate name, columns and primary key.

To create VIEW, insert manually its name and SELECT, which matches with the column order, into SQL request.

```
CREATE OR REPLACE VIEW VW_"Name of table" (
...
names of columns
...
) AS
```

The request for creating table need not be inserted. Its syntax is following.

```
CREATE TABLE "Name of table" (
...
names of columns, remapped D2000 types, NULL/NOT NULL
...
)

ALTER TABLE  "Name of table"
  ADD CONSTRAINT  pk_"Name of table" PRIMARY KEY (
    names of columns, whose field "Key" is marked
 )
```

The mapping of D2000 types to Oracle database types is stated below in the table.

D2000	Oracle
logical	NUMBER(1,0)
integer	NUMBER
analog	BINARY_DOUBLE
absolute time	DATE
relative time	BINARY_DOUBLE
text	VARCHAR2(256)

History depth - column

Name of a column of *Absolute time* type in the database. If the column name is entered, corresponding process D2000 DBManager will automatically delete all the rows, values of which in specified column is older than given history depth.

History depth - Months, Days, Hours

Time definition of the history depth.

History depth - Data purpose

Alternative form of history depth definition using *Data purpose* object type.

Related pages:

Databases and Database tables

Data category - Configuration Dialog Box

Data purpose - Configuration Dialog Box