D2000 DBManager

The process allows the D2000 real-time information system to access data stored in external database sources. It is a basic element for solving problems of vertical integration of information systems in enterprises.

The process **D2000 DBManager** provides access to external database sources by means of standard ODBC drivers of individual database systems. It also allows reading and writing data with access control.

The process **D2000 DBManager** was tested and should work with these databases:

- **Oracle Database** (versions 9i, 10g, 11g, 12c, 18c, 19c)
  *Note***:** We recommend applying available patches of the Oracle client as well as the Oracle ODBC driver - older versions of the Oracle client/driver suffered from various errors (handle leaks, instability, etc.).
- **Microsoft SQL Server** (versions 2000 - 2005)
  Note: in a specific situation, the UPDATE trigger was able to affect the DB(S)_UPDATE action so that it failed because the ODBC function SQLRowCount returned 0 (even though the data was updated). This case can be diagnosed by turning on the debugging category RTM. DBMANAGER.UPDATE.MSSQL.0 (if the category is active, it is ignored that SQLRowCount returns 0 changed rows).
- **Microsoft Access** (versions 2000 - XP)
- **PostgreSQL** (versions 9.x and above). D2000 Dbmanager supports the /NQ startup parameter to better support PostgreSQL (case sensitivity in table and column names).
  Note 1: in the configuration of ODBC drivers for PostgreSQL, we recommend setting the "Level of rollback on errors" parameter to *Statement*. The *Nop* setting makes the database connection unusable after any error. The *Transaction* setting will cause some actions to behave non-standardly (e.g. a DB(S)_INSUPD action inserting several rows will not insert anything if an error occurs while inserting any of the rows).
  Note 2: in the ODBC driver configuration for PostgreSQL, we recommend setting the "Max Varchar" parameter to a large enough value (larger than the maximum size of VARCHAR columns), otherwise the data may be truncated during page access. Similarly, we recommend setting the "Max LongVarchar" parameter to a sufficiently large value (larger than the maximum VARCHAR data size used in columns whose size is not explicitly specified).
  Note 3: in the configuration of ODBC drivers for PostgreSQL, we recommend leaving the "bytea as LO" parameter checked for correct work with BLOBs (DB_READ_BLOB, DB_UPDATE_BLOB).
- **Sybase Anywhere** (versions 6 - 12)
- **Informix** (driver version 4.10.00.16554, check a *Scrollable Cursors* option on the Advanced tab and set *Cursor Behavior* to value "1 - Preserve" on the Environment tab)
- **Firebird** (version 2.0.4, native ODBC driver or driver from GM - Software)
  *Note 1***:** Native ODBC driver (version 2.0.0.148) doesn't enable editing in paged mode (browser in HI, actions PG*). Seemingly editing works but no data are modified in the database.
  *Note 2***:** ODBC driver from GM - Software supports editing and inserting rows in paged mode, but it displays only the first row of the table.

Note: Starting with D2000 version 11.2.57, patches from January 14, 2019, and later, a startup parameter /DBS was implemented to enable support for long strings (> 4000 characters).

Note: since D2000 v22.0.74, there is protection against *SQL injection* attacks in the **D2000 DBManager** process. The protection generates the error code "Malevolent SQL command detected" (error no. 666). Protection can be turned off with the start parameter /E-DBG.DBMANAGER.SANITIZE or at runtime using the D2000 System Console by disabling the DBG.DBMANAGER.SANITIZE category.

From version D2000 v7.0, along with the "standard" process **D2000 DBManager,** a variant that only works with Oracle databases (**dbmanager_ora.exe**) is implemented. This variant does not use the ODBC interface but accesses the database via the OCI (Oracle Client Interface). It is available on Windows platforms as well as on OpenVMS. When using the **dbmanager_ora.exe**, the configuration dialog box of an object of Database type contains TNS (Transparent Network Substrate) in the parameter DSN (see the description of the file ***tnsnames.ora*** in the Oracle documentation).

Advantages of the use of **dbmanager_ora.exe** compared to **dbmanager.exe**:

- Availability on OpenVMS and HPUX platforms.
- Possibility of page access to index-organized tables of Oracle database (IOT- index-organized tables). ODBC drivers Oracle contain an error (tested for versions 9.02.00.00 up to 9.02.00.06), that causes an application crash on the page access to IOT and therefore the ODBC version of the DBManager does not provide page access if it finds out that the table is IOT.
- Less leakage of system resources: ODBC drivers for Oracle (tested for versions 9.02.00.00 up to 9.02.00.06) leak 4 handles in a connect /disconnect cycle, OCI interface leaks only one :-) .. in any case, we recommend setting a number of prepared connections for the object of Datab ase type larger enough in order to prevent frequent opening and closing of the connections.
- Possibility to enable the debug information on the status of the Oracle cluster (see the parameter /DBCY).

Various runtime categories of the **dbmanager.exe** process:

- **RTM.DBMANAGER.UPSERT** - On the PostgreSQL platform when inserting data with the DB(S)_INSUPD action, UPSERT (INSERT ... ON CONFLICT (...) DO UPDATE) is used. This syntax is supported by PostgreSQL since version 9.5.
- **RTM.DBMANAGER.UPDATE.MSSQL.0** - On the Microsoft SQL Server when updating data with DB(S)_UPDATE, DB(S)_INSUPD actions, an erroneously returned 0 affected rows (as a result of the UPDATE trigger on a table) is ignored.

Different runtime categories of the **dbmanager_ora.exe** process:

- **RTM.DBMANAGER.STRICT_CHECK** - If there is a column of text type in the table and this column is numeric in D2000, an error in converting to a number may occur during reading (*ORA_01722: invalid number*). If this runtime category is active, a strict check of column types is performed, and if such a column is detected, DbManager will refuse to work with the table (and the log will contain error messages).
- **RTM.DBMANAGER.NO_COUNT_OVER** - During page access (PG_CONNECT action and a browser), by default, the number of records is determined within the data-reading SQL query (*COUNT(*) OVER ()*). In some cases, this reading can be slower than running an explicit query (*S ELECT COUNT(*) FROM <table>*). We noticed a specific case on Oracle 19.21 and a table with more than 2.6 million rows (at an SGA size of 12GB). If this runtime category is active, the number of rows will be determined using an explicit query before the data is read.

Blog

You can read the following blogs about DbManager:

- D2000 and working with databases

Related pages:

System processes