Upgrade of application to D2000 v9.1

In D2000 System since version 9.1, a support of Sybase database platform has been changed. With regard to support of unicode strings, we had to start using Sybase 12 instead of older Sybase 9 in some application database. This implies that any application, which will use Sybase database platform, must contains both its configuration database (syscfg.db) and log database (logfile.db) in the version Sybase 12.

When creating a new application in D2000 System v9.1, there are not necessary any additional steps for support of Sybase 12, just creating an application in a standard way. The option for creating the application on Sybase 9 platform is disabled.

When porting an old application from the version lower than D2000 v9.1, there is necessary to upgrade the application database. The upgrade converts the selected databases from Sybase 9 to Sybase 12. Therefore, on the destination workstation it should be installed Sybase 12 and created at least one engine on this platform. Sybase 9 need not be installed.

You should realize that time necessary for upgrading the database depends on its size on a disc. If the database is large (>2 GB), upgrading can last even several hours.
For porting, you should do these steps:

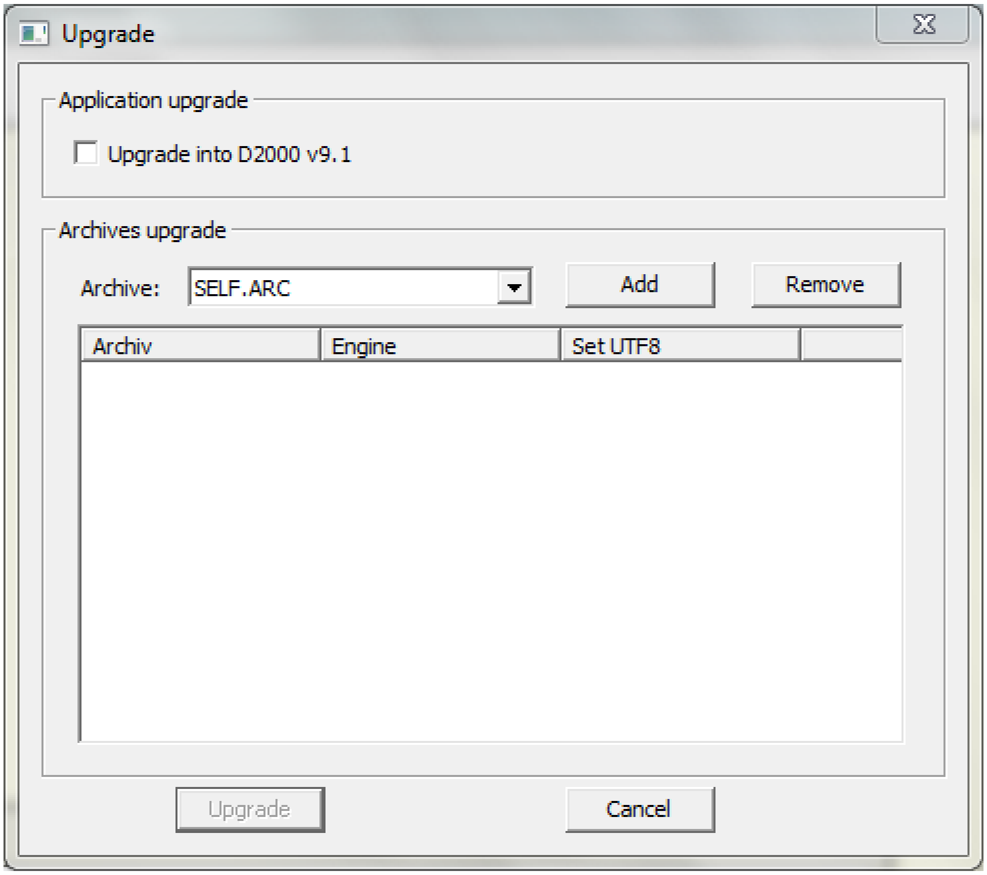
- In D2000 v9.1 and higher create a new application in the module D2000 Application Manager.
- Copy the application databases from the directory structure of old application (version lower than D2000 v9.1):
    - SysCfg.db
    - LogFile.db – if the size of log database is too large (upgrading of application could take several hours), in some situations is better to replace this file with an empty database, which is placed in the directory templates\Sybase9 in the D2000 System installation directory.
    - Archiv.db
- Perform the automatic upgrade described in the next paragraphs.

Note: From the version D2000 v9.1, Sybase 9 is still supported for the archive database. Therefore, the databases need not be upgraded, the application will work even in the newer versions. If an user decides to keep the archive databases on Sybase 9, he must create a corresponding engine on Sybase 9. It means, that also Sybase 9 must be installed on the destination workstation. The automatic upgrading of the application does not influence the upgrade of depository databases. These databases must be upgraded manually according to these steps.
From the version Sybase 12, the support of compressed depository databases with the suffix .cdb was cancelled. Therefore, if such database is to be upgraded to Sybase 12, the compressed database must be changed into non-compressed with the suffix .db and then to be upgraded.

Upgrade

Choose the item Applications, which shows the list of applications, in the dialog window **D2000 Enterprise Manager**, tab **Applications.** After right-clicking over the particular application, the popup menu opens to upgrade the application. This item is disabled if the application is running, i.e. database can be upgraded only if the application is not running.

When selecting the item **Upgrade into D2000 v9.1**, this dialog window opens:



Archive: SELF.ARC

This dialog window consists of two parts:

1. Application upgrade
2. Archives upgrade

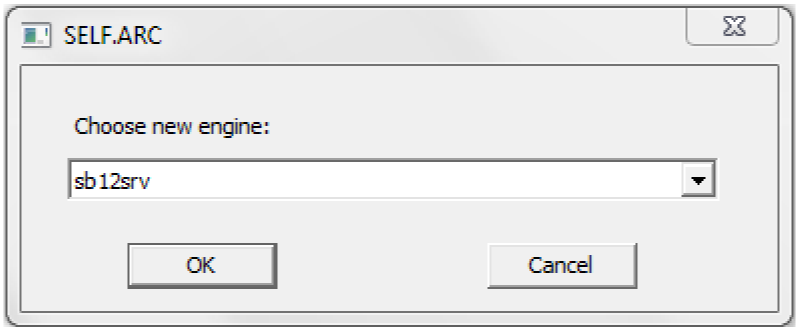
Application upgrade

This part enables to set the upgrade of application to D2000 v9.1. This step involves the conversion of databases to Sybase 12.

The application upgrade means, in fact, the upgrading of **SysCfg.db** and **LogFile.db**, which are in each application. If the conversion fails because of any reason, the databases will be automatically backed-up (see the activities when upgrading). When upgrading, the encoding of database will be changed. The UTF-8 encoding is set for the strings.

Archives upgrade

This part contains the list of application archives that can be upgraded. Select the particular archive in the drop-down list. After clicking on the button **Add**, a new dialog box opens to choose a new engine.



SELF.ARC

Choose new engine: sb12srv

After clicking on **OK** button, the archive is automatically included into the list of upgrades. It can be removed from the list by the button **Remove**.
The list cannot contain the same archives. If the particular archive is in the list, first it must be removed and then you can change the engine or encoding.

The process of upgrading is executed after clicking on **Upgrade** button. The upgrade of application (if chosen) is done at first and then each archive that is in the list is upgraded.

Note: This activity may take quite long and fully use the power supplies of a computer depending on the size and number of upgraded databases. If any of the databases is too big, you can replace it manually, if the circumstances permits it, by a new empty template (not for syscfg.db). Its upgrading takes less time.

The following activities are done when upgrading each database:

- An auxiliary folder *conversion* is created in the directory, which contains the database. A backup file, whose name consists of the database name and the creation time, is created here. This folder also contains various auxiliary files that are useful for database upgrade. After the upgrade is done successfully, these files are unnecessary and can be removed.
- A new Sybase 12 database is created in the folder *convert*. All data are moved here from the previous database.
- Old database is replaced by the new one and the file, containing the new database, in the auxiliary folder is deleted. The old database backup remains.

The auxiliary folders and copy of database are created if only the change of database encoding (no upgrade) is required.

Upgrading a database

(do not use for upgrading syscfg.db, logfile.db, archiv.db)

The upgrade requires Sybase 12 (Sybase 9 is not required for upgrade to a higher version). Check, if the system variable *%SqlAny12%* points to the path in the file system in which Sybase 12 is installed (mostly it is c:\Program Files\SQL Anywhere 12\). The utilities - dbunload, dbinit, dbisql, which are placed in the folder **BinXX** (XX = 32/64, depending on whether the application is upgraded to 32 or 64 bit version), are used.

For upgrading the database, you may use the following script corresponding with the version of installed Sybase 12 on the destination workstation.

For 32-bit Sybase 12: upgrade.cmd
For 64-bit Sybase 12: upgrade_win64.cmd

How to use the script:

1. Open cmdline in the directory with database.

2. Start script as per the installed version of sybase (upgrade/upgrade_win64) as following:

   upgrade.cmd <database_name> \[<db_user>\] \[<db_password>\]

database_name	- name of database (or full path)
db_user	- name of database user - if it is not entered, default dba will be used
db_password	- password of database user - if it is not entered, default swx will be used

3. Wait for the result that will be written up at the end of executing the script.

Manual upgrading database

If the user wants fully control the upgrade of database, the following procedure may be applied. It is recommended to make a backup of databases before upgrading (copy *.db file into other folder), in case that the upgrade could fail.

The upgrade is done in five steps:

1. The file *authenticate.sql*, which is placed in the subdirectory **Templates** in the installation directory D2000, is copied into the directory *%SqlAny12%/scripts.*
2. The database is disconnected. A reload sql script, which is able to re-create the all database, is created.

   ```
   "%SqlAny12%\BinXX\dbunload" -c "UID=dba;PWD=swx;DBF=dbPath" dbPathDir -r reloadPath -o logPath
   ```

3. The new database is created in the version 12.

   ```
   "%SqlAny12%\BinXX\dbinit" -p 16384 -z utf8 emptydbPath
   ```

   Note: The value 16384 represents a page size in database, which is used by D2000 System for Sybase12. However, it is not obligatory. The encoding, which is used for database, is placed behind parameter *-z.*
4. Data are copied into the new database by the reload script, which has been generated in the first step.

   ```
   "%SqlAny12%\BinXX\dbisql" -c "uid=dba;pwd=sql;dbf=emptydbPath" -nogui READ reloadPath
   ```

5. The original database is replaced by new one.

   ```
   COPY emptydbPath  dbPath
   ```

Explanations:

BinXX	Possible values: Bin32, Bin64
dbPath	The path to the database file that will be upgraded to Sybase12. The path must consist of the database name together with the extension. Example: C:\D2000\D2000.APP\mojapokusna\SELF1\Archiv.db
dbPathDir	The path to directory which contains a database file. It must be just path to directory without a back slash '\'. It must not contain the name of database file. Example: C:\D2000\D2000.APP\mojapokusna\SELF1
reloadPath	The path in the directory structure, in which the reload script (including its name), is created. Example: C:\D2000\reload.sql
logPath	The path in the directory structure, in which the log file is created. This file contains all information provided by *dbunload* utility. Example: C:\D2000\dbunload.log
emptydbPath	The path in the directory structure, in which the new database is created, including the name and extension. It should not be the same as dbPath. Example: C:\D2000\D2000.APP\mojapokusna\SELF1\NovyArchiv.db

Note: The path to the directory or files, which contains a blank space, must be enclosed in the quotes.

If all steps are completed successfully, the database is upgraded in the folder dbPath. When upgrading, the database creates several files with .dat extension and one reload script in the folder *reloadPath*. They are unnecessary after successful upgrade.

ⓘ **Related pages:**

Management of D2000 system applications