



# ProjectWorX32 – Unpacking SQL Databases

## APPLICATIONS NOTE

August 2004



**Description:** This Application Note is meant for all users who are working with ProjectWorX32 and integrated Alarm Logging or Trend Logging configurations logging to SQL/MSDE databases. When ProjectWorX32 unpacks a project on a target PC, the unpacking engine typically recreates all related DSN (Data Source Names) and logging databases for the Alarm and Trend Loggers on the fly. However, this applies only to MS Access logging databases in version 7.0. ProjectWorX32 v7.0 will not be able to recreate a SQL/MSDE database on the fly. The user will have to create the SQL/MSDE database and the DSN manually. ICONICS is working to resolve this issue with the next version of GENESIS32 7.x. This Application Note will explain how to work around the issue

**OS Requirement:** Windows XP/2000/NT/98 SE.

**General Requirement:** GENESIS32 and SQL Server/MSDE installed on the target PC on which the project is going to be unpacked.

### Assumptions:

We assume that you already have a TrendWorX32 or AlarmWorX32 configuration, which are configured to log to a SQL/MSDE database. You have already packed the project on a source PC (the PC on which the project originated). You are trying to unpack your project on the target PC, which has GENESIS32 and SQL Server/MSDE installed.

### Steps:

You will now go through the steps of recreating the SQL database and DSN on the target PC. We will explain the manual procedure based on the TrendWorX32 Configurator. The same procedure also applies to the Alarm Logger Configurator.

1. Unpack your project and make the **project active**. You will notice that the Configuration database is also made active automatically.
2. **Locate your TrendWorX32 Logging configuration in ProjectWorX32.** Double-Click on the Configuration database to open it. Go to the Database group and check for the **DSN name (ODBC Data Source)** here. Make a note of the DSN name and close the window. See Figure 2 below:

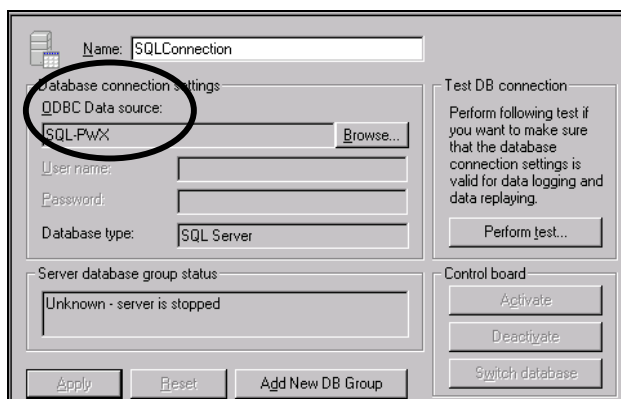


Figure 2

3. Now open the **Control Panel** → **Administrative Tools** → **Data Sources (ODBC)** to find the name of the associated SQL/MSDE database. Click on the **System DSN** tab. The DSN is already listed here. See the figure below:

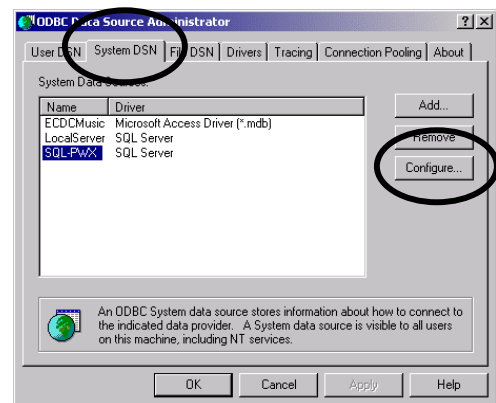


Figure 3

4. Click on the **Configure** button and the **Microsoft SQL Server DSN Configuration** window will be opened as shown in the figure below. The DSN name is already listed here. See the figure below:

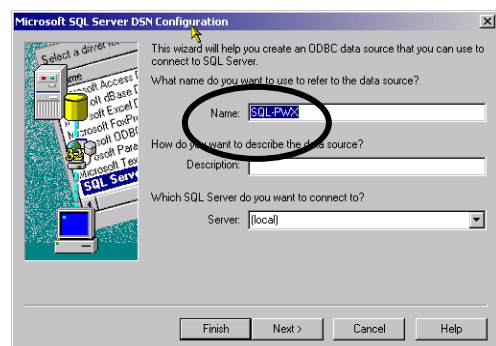


Figure 4

5. Click on **Next** until you see the following user interface (Figure 5). In this window the actual SQL/MSDE database name used in your project is listed. We will have to create this database manually since there is no database physically present on the target PC yet. **Copy** (Ctrl+C) the name exactly as you see it and **close** the window. See the figure below:

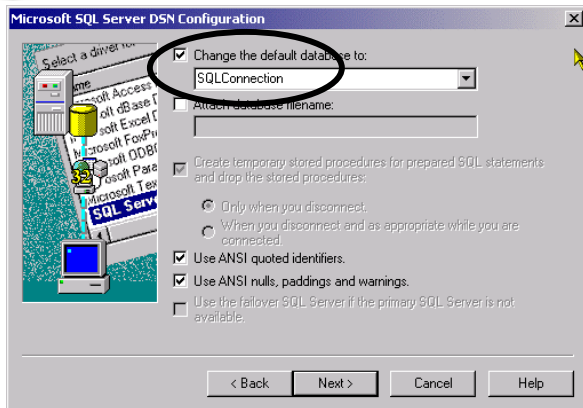


Figure 5

6. We will create a Database with exactly the same name. Go to **Start – Programs – ICONICS Genesis32 – Tools – MSDE Manager** and click to **Open** it.
7. The **MSDE manager** will open as shown in the figure below. You will notice that you do not have any Database with the name as shown in Figure 5. See the figure below:

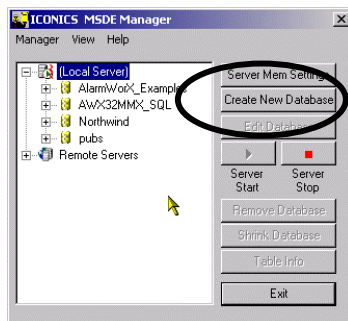


Figure 7

8. Click on **Create New Database** button, to create a new Database. For the database name, paste the name, which you have copied from Step 5. The location of the database is up to the user at this point. Click on **OK**, to create the database and close the window. See the figure below:

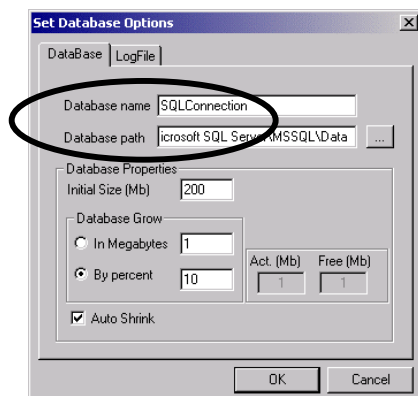


Figure 8

9. The **Database** will be created as shown in the figure below:

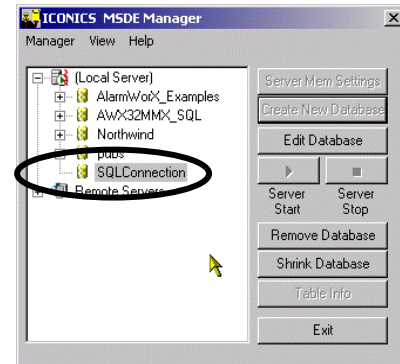


Figure 8

Don't forget to change the memory settings for SQL Server/MSDE. SQL Server/MSDE should not consume more the 50% of the physical RAM. You may close the ICONICS MSDE Manager now.

10. This should be all you have to do. We basically recreated the SQL/MSDE database on the target PC manually.