Installing and Configuring Database Access

This section describes all steps required to run a service that connects to databases:

- Installing SQL client tools
- Defining the SQL adapter preferences
- Defining database parameters

When installing the BRIDGE, all additional libraries (SQL Libraries, SAP Libraries, and Java Libraries) are being installed as well.
After a successful installation, the BRIDGE is capable to connect to database backends.

First, you need to install the client tools or the ODBC drivers and configure each database access if needed.
Databases like DB2 or Oracle are usually running on a dedicated database server. The BRIDGE opens a connection to the database server via the database client tools or ODBC.

On this Page:

Related Pages:

- Installing and Configuring Database Access for DB2
- Installing and Configuring Database Access for Informix
- Installing and Configuring Database Access for Microsoft SQL Server
- Installing and Configuring Database Access for MySQL
- Installing and Configuring Database Access for Oracle
- Troubleshooting Database Access
- Installation Modes and BRIDGE Domain

Related Documentation:

- Database-Specific Mappings
- xUML Service Settings

Prerequisites

The BRIDGE has been successfully installed. When running the BRIDGE in server mode (as opposed to workstation mode), the BRIDGE node instance has to be a member of a BRIDGE domain. For more information on installation modes and BRIDGE domains, refer to Installation Modes and BRIDGE Domain.

Supported Databases

Database	Versions
DB2	DB2 CLI version 6.x or higher
Informix	Informix CLI version 2.x or higher
InterBase	InterBase client API version 5.x or higher, all Firebird versions
Microsoft SQL Server	**ODBC:** SQLServer ODBC version 2000.x or SQL Server Native Client (ODBC API) version 2005 and higher
	OLE DB: SQL Server OLE DB version 2.5 or higher
MySQL	MySQL C API version 3.23.x or higher
Oracle	Oracle OCI version 8 or higher
PostgreSQL	libpq version 7.1.x or higher
SQLBase	CAPI version 6.x or higher
SQLite	libsqlite version 3.x or higher
Sybase	Open Client version 10.0 or higher

Installing Database Access

- To install database access to a DB2 database refer to Installing and Configuring Database Access for DB2.
- To install database access to an Oracle database refer to Installing and Configuring Database Access for Oracle.
- To install database access to a MySQL database refer to Installing and Configuring Database Access for MySQL.

If you want to access the installed database via the xUML RUNTIME that is embedded in **Scheer PAS** *BUILDER*, please note that you have to make the database client available to the BUILDER:

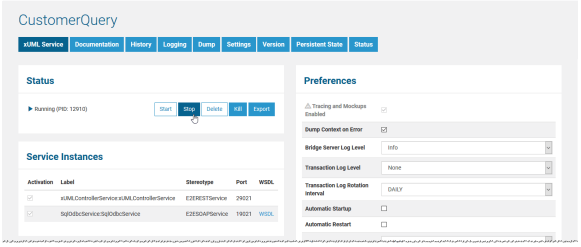
- Install a database client to the system you are running the BUILDER on.
- Make sure that the database client location makes part of the PATH variable.
  After having changed PATH, you need to restart MagicDraw to make it read the new content of PATH.

Defining the ODBC Database Parameters

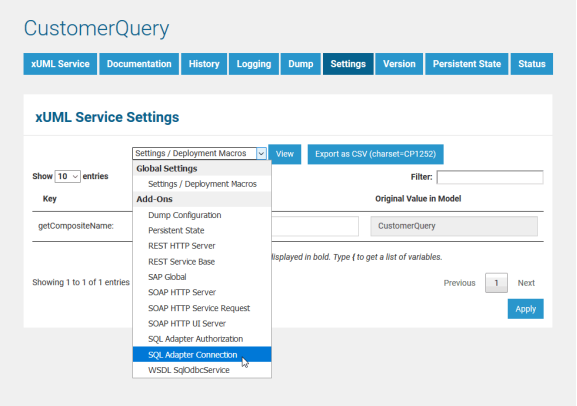
All well-known databases like Oracle, DB2, Inter Base, Firebird, dBase, Access, FoxPro, Paradox, SQL Server, and so on can be connected via ODBC.
Install the required ODBC drivers and define the parameters for each connection.

Before starting a deployed xUML service that connects to a database backend, you may want to redefine required database parameters in the BRIDGE.
You find more information about database-specific information (database names and mappings) in section Database-Specific Mappings.

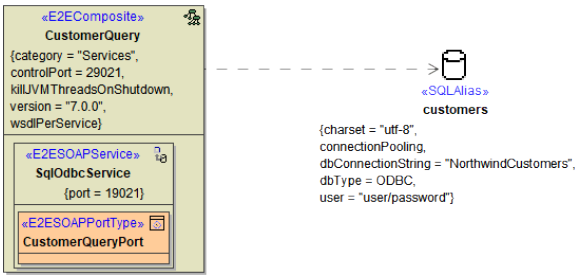
Select the xUML service in the navigation (in this example **CustomerQuery** and stop the service.

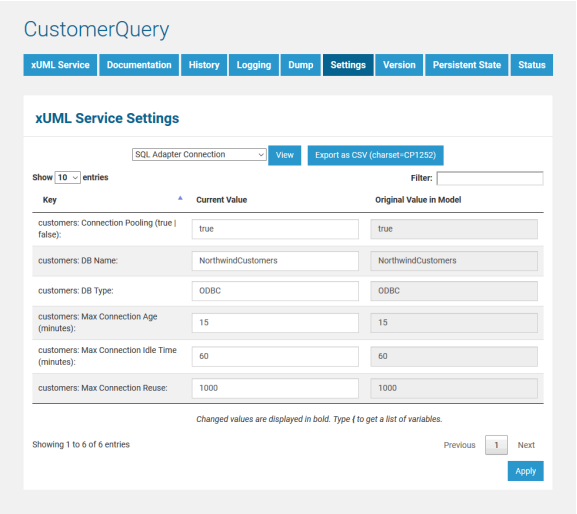


Switch to tab **Settings**. From the dropdown list, select the option **SQL Adapter Connection** in the **Add-ons** category.



The default parameters are defined in the component diagram of the xUML service (see CustomerQuery example below). The connection is defined by a dependency between the composite named **CustomerQuery** and the SQL alias **customers**. The database user and password are defined in the tagged value **user** on the SQL alias.



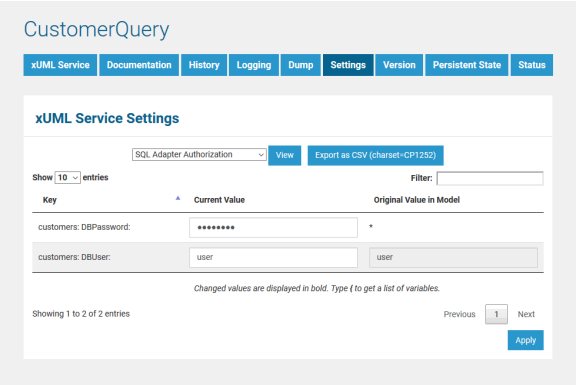


Enter the following database parameters and click **Apply**:

Key	Value
customers: DBType	Name of the SQL Service component e.g. **ODBC**
customers: DBName	Name of the database, e.g. **NorthwindCustomers**
customers: Connection Pooling	Added in Builder 5.1.8.58 Runtime 5.1.82.0 This tagged value controls the connection pooling. If true, each connection is put into a pool after use. If an SQL adapter requires a connection, it is taken from the pool. If no connection is available, a new connection is being created and put into the pool after use. The time the connection is kept in the pool depends on the other pooling parameters.
customers: Max Connection Reuse	This tagged value controls how often a connection can be re-used. After the connection has been re-used for **maxConnectionReuse**, it will be closed and not put back into the pool. This feature has been introduced because some databases had problems if the connection was re-used too often. Value **-1** means the connection will be re-used forever. In this case you should define reasonable values for **maxConnectionAge** or **maxConnectionIdleTime** (see above). Note that the pooling is implicitly switched off, if **maxConnectionReuse** is set to 0.
customers: Max Connection Age	After a given connection age (in minutes) the connection will be closed and removed from the pool.
customers: Max Connection Idle Time	Connections not used for the time specified (in minutes) will be closed and removed from the pool. This is useful for connections going through firewalls because such connections might be cut off after some time.

For each SQL adapter alias found in the activity diagrams of a UML model, you will find a pair of the two parameters **DBType** and **DBName** on the SQL adapter connection settings. According to the example with the both parameters in the table above, you would find an alias called customers in the UML model of the deployed xUML service CustomerQuery.

Now, select **SQL Adapter Authorization** from the list.



Enter the following database parameters and click **Apply**:

Key	Value
customers: DBUser	Database user
customers: DBPassword	Database password

Users who do not have the permission to view the settings cannot see the **Settings** tab at all. This prevents that unauthorized users access sensitive information like passwords, etc.

For more details on changing the settings refer to xUML Service Settings.