

# SQL



This page explains the **SQL Adapter** in Bridge context. If you were looking for the same information regarding the [PAS Designer](#), refer to [SQL Adapter](#) in the Designer guide.

## Tagged Values

### <<SQL Alias>>

| Tagged Value                     | Description   | Allowed Values   |
|----------------------------------|---|--|
| General                          |   |  |
| <b>dbConnectionString</b>        | The format of the database connection string depends on the type of the database. For more details see <a href="#">Database Server-Specific Notes for SQL Adapters</a> .  |  |
| <b>dbType</b>                    | Type of the database.   | Oracle, SQLServer, InterBase, SQLBase, ODBC, DB2, Informix, Sybase, MySQL, PostgreSQL, SQLite, DbTypeVariable  |
| <b>dbTypeVariable</b>            | <p>If the tagged value <b>dbType</b> is set to <b>DbTypeValue</b>, the <b>dbTypeVariable</b> tagged value is used to define the type of the database. The <b>dbType</b> then can be defined by a setting variable.</p> <p>This is to handle the case, that you not want to hard code the <b>dbType</b>, but to configure it at runtime via the E2E Bridge.</p> <p>See <a href="#">Using Global Setting Variables</a> for more information on how to define a global setting variable in the E2E Bridge.</p> | <p>Any global setting variable from the E2E Bridge.</p> <p>Example: <code>{{my_setting_variable}}</code></p>   |
| <b>user</b>                      | DB user. Optional the password can be given after a '/'. However, this is recommended for development purposes only.  | Example: <code>{{DB_USER}} / {{DB_PASSWORD}}</code>  |
| <b>options</b>                   | This tagged value can hold a comma separated list of <name>=<value> pairs. These list elements are interpreted as native options. The possible name-value pairs depend on the database type. A comprehensive list can be found at <a href="https://www.sqlapi.com/ApiDoc/servers/">https://www.sqlapi.com/ApiDoc/servers/</a>   | Example: <code>SSPROP_INIT_ENCRYPT=VARIANT_TRUE</code>   |
| <b>transactionIsolationLevel</b> | <p>Bridge 7 Specify here the required transaction isolation level of the SQL connection according to SQL-92 standard. Refer to <a href="#">Wikipedia</a> for a detailed description of the available isolation levels.</p> <p>Please note that not all databases support all levels. In this case a database-specific mapping will occur.</p> <p>For persistent state databases no other than <b>&lt;UNSPECIFIED&gt;</b> and <b>DB MS default</b> are allowed.</p>  | <p><b>Default</b> Use the default isolation level of the connected database system.</p> <p><b>Read Dirty</b> Lowest isolation level. Dirty reads allowed, SQL adapter may fetch not-yet-committed changes of other transactions.</p> |

#### On this Page:

- [Tagged Values](#)
  - [<<SQL Alias>>](#)
  - [<<SQL Adapter>>](#)
  - [<<SQLConfigurationAdapter>>](#)
- [SQL Adapter Parameters](#)
- [SQL Configuration Adapter Parameters](#)
- [SQL Adapter Parameter Types](#)
  - [SQLConfiguration](#)
  - [SQLAuthentication](#)

#### Related Pages:

- [Database Server-Specific Notes for SQL Adapters](#)
- [Database-Specific Mappings](#)
- [Charset Definitions](#)
- [Time Zones](#)
- [Using Global Setting Variables](#)
- [SQL Options Documentation](#)
- [Troubleshooting the SQL Adapter](#)
- [Database Isolation Levels \(Wikipedia\)](#)

|                    |  |  |  |
|--------------------|--|--|--|
|                    |  | R<br>e<br>a<br>d<br>c<br>o<br>m<br>m<br>i<br>t<br>t<br>e<br>d  | Lock-based concurrency control.  |
|                    |  | R<br>e<br>p<br>e<br>a<br>t<br>a<br>b<br>l<br>e<br>r<br>e<br>a<br>d   | Lock-based concurrency control.  |
|                    |  | S<br>e<br>r<br>i<br>a<br>l<br>i<br>z<br>a<br>b<br>l<br>e   | Highest isolation level. Lock-based concurrency control.   |
| Localization       |  |  |  |
| charset            | Any database uses a charset to encode Strings. If the database uses UNICODE charsets (UTF-8, UTF-16, UTF-32), encoding is handled automatically. If the database is not UNICODE compliant, the Bridge assumes 7-bit ASCII by default. However, in many cases it necessary to define the charset explicitly. This is done by the tagged value <b>charset</b> as shown below. The charset needs to be the same as defined at the database settings. All possible charset definitions are listed in section <a href="#">Charset Definitions</a> . | Example: UTF-8<br>See <a href="#">Charset Definitions</a> for a list of possible values.                     |  |
| timezone           | You can enter a valid time zone or the value <b>local</b> , which uses the time zone of the xUML service. See <a href="#">Time Zones</a> for a list of possible values. If <b>timezone</b> does not contain any content (is NULL), UTC is used.  | D<br>e<br>f<br>a<br>u<br>l<br>t<br>i<br>s<br>N<br>U<br>L<br>L  | Example: "Australia/Melbourne", "CET", "Etc/GMT+10"  |
| unicodeMode        | Added in Builder 6.0.15.5 Runtime 2015.15 Specify the encoding for database access.<br><br><div>We recommend to use the <b>Platform default</b> unless you suspect an encoding incompatibility (see <a href="#">Troubleshooting the SQL Adapter</a>). This option represents the former behavior and is fully backwards-compatible - means, it can be used with older xUML Runtimes. The two other (force mode) options will be ignored by older Runtimes without warning.</div>   | P<br>l<br>a<br>t<br>f<br>o<br>r<br>m<br>d<br>e<br>f<br>a<br>u<br>l<br>t<br>(d<br>e<br>f<br>a<br>u<br>l<br>t) | Use the platform default mode. This is <ul style="list-style-type: none"><li>Unicode: for Windows systems</li><li>non-Unicode: for all others</li></ul> This option is backwards compatible to older Runtimes. |
|                    |  | U<br>n<br>i<br>c<br>o<br>d<br>e  | Force Unicode mode.  |
|                    |  | n<br>o<br>n<br>-<br>U<br>n<br>i<br>c<br>o<br>d<br>e  | Force non-Unicode mode.  |
| Connection Pooling |  |  |  |
| connectionPooling  | 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  | tr<br>ue   | Database connections are pooled.   |

|  |  |  |  |
|--|--|--|--|
|  | use. The time the connection is kept in the pool depends on the other pooling parameters.  | false Database connections are not pooled.   |  |
| maxConnectionAge   | After a given connection age (in minutes) the connection will be closed and removed from the pool.   | Connection age in minutes, default is <b>15 minutes</b> , <b>-1</b> means forever. |  |
| maxConnectionIdleTime  | Connections not used for the time specified (in minutes) will be closed and removed from the pool.<br>This is useful for connections going through firewalls because such connections might be cut off after some time.  | Values in minutes, default is <b>60</b> .  |  |
| maxConnectionReuse   | This tagged value controls how often a connection can be re-used. After the connection has been re-used for <b>maxConnectionReuse</b> , 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 <b>-1</b> means the connection will be re-used forever. In this case you should define reasonable values for <b>maxConnectionAge</b> or <b>maxConnectionIdleTime</b> (see above).  | 0  | pooling is implicitly switched off.                  |
|  |  | -1   | connections are pooled forever                       |
|  |  | average  | number of connections to be pooled, default is 1000. |
| <div>Note that the pooling is implicitly switched off, if <b>maxConnectionReuse</b> is set to 0.</div> |  |  |  |
| Qualifier  |  |  |  |
| schema   | String that prefixes tables and stored procedures. For example, if schema is set to S1, all tables accessing the current DB are prefixed by "S1".<br><div>This works only if the tables are marked using the <code>TABLE::</code> keyword, e.g. <code>TABLE::EMPLOYEE</code> in SQL statements. If you do not prefix the table name by <code>TABLE::</code>, the tablename is used as it is.</div>   |  |  |
| tableQualifier   | String that prefixes tables. For example, if <b>tableQualifier</b> is set to TQ1, all tables accessing the current DB are prefixed by "TQ1", e.g. <code>TQ1EMPLOYEE</code> . If schema and table qualifier are given, all tables will become: <code>&lt;schema&gt;.&lt;tableQualifier&gt;&lt;tableName&gt;</code> .<br><div>This works only if the tables are marked using the <code>TABLE::</code> keyword, e.g. <code>TABLE::EMPLOYEE</code> in SQL statements. If you do not prefix the table name by <code>TABLE::</code>, the tablename is used as it is.</div> |  |  |

## <<SQL Adapter>>

| Tagged Value             | Description  | Allowed Values                       |  |
|--------------------------|--|--------------------------------------|--|
| <b>alias</b>             | Specify the SQL alias resp. the database the adapter should connect to.  | any valid SQL alias                  |  |
| <b>action</b>            | Holds the action to perform on the database.   | <b>execute</b> (default)             | Execute an SQL statement.  |
|                          |  | getHandle                            | Get a connection handle for subsequent <b>fetchNext</b> actions.                                   |
|                          |  | fetchNext                            | Fetch next record.   |
|                          |  | closeHandle                          | Close the connection handle. If all records have been fetched, the handle is closed automatically. |
| <b>dbType</b>            | Overwrite the database type defined in the SQL alias.  |                                      |  |
| <b>prefetchedRecords</b> | Number of pre-fetched records. This makes sense for SQL queries, especially when using <b>getHandle</b> / <b>fetchNext</b> actions for bulk fetch use cases. |                                      |  |
| <b>sql</b>               | Holds the SQL statement to be performed on the database.   | Any valid SQL statement as a string. |  |

## <<SQLConfigurationAdapter>>

| Tagged Value | Description   | Allowed Values              |                                       |
|--------------|---|-----------------------------|---------------------------------------|
| alias        | Specify the SQL alias resp. the database to configure | any valid SQL alias         |                                       |
| action       | Specify the configuration action.                     | setAuthentication (default) | Set the authentication configuration. |

## SQL Adapter Parameters

| Name          | Type      | Direction | Description  |
|---------------|-----------|-----------|--|
| sql           | String    | in        | Use this parameter to provide a dynamic SQL statement.   |
| inputBindings | Map       | in        | Use this parameter to provide parameter/value pairs for parameterized statements.  |
| affectedRows  | Integer   | out       | This parameter returns the number of rows affected by the SQL statement.   |
|               | any class | out       | Result set of the database query.<br>The SQL Adapter tries to match the table column names with the attribute names of the output class. For information on type mapping refer to <a href="#">Database-Specific Mappings</a> . |

## SQL Configuration Adapter Parameters

| Name          | Type                             | Direction | Description  |
|---------------|----------------------------------|-----------|--|
| configuration | <a href="#">SQLConfiguration</a> | in        | Use this parameter to provide SQL connection configurations. At the moment these are authentication configurations only. |

## SQL Adapter Parameter Types

### SQLConfiguration

| Attribute      | Type                              | Description                                     | Values/Example |
|----------------|-----------------------------------|---|----------------|
| authentication | <a href="#">SQLAuthentication</a> | An object containing the user and the password. |                |

### SQLAuthentication

| Attribute | Type   | Description | Values/Example |
|-----------|--------|-------------|----------------|
| password  | String | Password.   |                |
| username  | String | Username.   |                |