DBLookup Mediator[¶](https://ei.docs.wso2.com/en/7.0.0/micro-integrator/references/mediators/dBLookup-Mediator/#dblookup-mediator)

The **DBLookup Mediator** can execute an arbitrary SQL select statement and then set a resulting values as a local message property in the message context. The DB connection used may be looked up from an external data source or specified inline.

**Info**

The DBLookup mediator is a [content-aware](https://ei.docs.wso2.com/en/7.0.0/micro-integrator/references/mediators/about-mediators/#classification-of-mediators) mediator.

The DBLookup mediator can set a property from one row in a result set. It cannot return multiple rows. If you need to get multiple records, or if you have a table with multiple parameters (such as URLs), you can create a data service and invoke that service from the Micro Integrator using the [Callout mediator](https://ei.docs.wso2.com/en/7.0.0/micro-integrator/references/mediators/callout-Mediator/) instead.

Syntax[¶](https://ei.docs.wso2.com/en/7.0.0/micro-integrator/references/mediators/dBLookup-Mediator/#syntax)

The syntax of the DBLookup mediator changes depending on whether you connect to the database using a connection pool, or using a data source. Click on the relevant tab to view the required syntax.

**Connection Pool**

<dblookup>

<connection>

<pool>

<driver/>

<url/>

<user/>

<password/>

<property name="name" value="value"/>\*

</pool>

</connection>

<statement>

<sql>select something from table where something\_else = ?</sql>

<parameter [value="" | expression=""] type="CHAR|VARCHAR|LONGVARCHAR|NUMERIC|DECIMAL|BIT|TINYINT|SMALLINT|INTEGER|BIGINT|REAL|FLOAT|DOUBLE|DATE|TIME|TIMESTAMP"/>\*

<result name="string" column="int|string"/>\*

</statement>+

</dblookup>

**Data source** The syntax of the DBLookup mediator further differs based on whether the connection to the database is made using an external datasource or a Carbon datasource.

External Datasource

<dblookup>

<connection>

<pool>

<dsName/>

<icClass/>

<url/>

<user/>

<password/>

<property name="name" value="value"/>\*

</pool>

</connection>

<statement>

<sql>select something from table where something\_else = ?</sql>

<parameter [value="" | expression=""] type="CHAR|VARCHAR|LONGVARCHAR|NUMERIC|DECIMAL|BIT|TINYINT|SMALLINT|INTEGER|BIGINT|REAL|FLOAT|DOUBLE|DATE|TIME|TIMESTAMP"/>\*

<result name="string" column="int|string"/>\*

</statement>+

</dblookup>

Carbon Datasource

Configurations[¶](https://ei.docs.wso2.com/en/7.0.0/micro-integrator/references/mediators/dBLookup-Mediator/#configurations)

The configuration of the DBLookup mediator changes depending on whether you connect to the database using a connection pool, or using a data source.

Connection Pool configurations[¶](https://ei.docs.wso2.com/en/7.0.0/micro-integrator/references/mediators/dBLookup-Mediator/#connection-pool-configurations)

The parameters available to configure the DBLookup mediator are as follows:

**Info**

When specifying the DB connection using a connection pool, other than specifying parameter values inline, you can also specify following parameter values of the connection information (i.e. Driver, URL, User and password) as registry entries. The advantage of specifying a parameter value as a registry entry is that the same connection information configurations can be used in different environments simply by changing the registry entry value. To do this, give the registry path within the key attribute as shown in the example below.

<**dblookup** xmlns="http://ws.apache.org/ns/synapse">

<**connection**>

<**pool**>

<**password** key="conf:/repository/esb/password"/>

<**driver** key="conf:/repository/esb/driver"/>

<**url** key="conf:/repository/esb/url"/>

<**user** key="conf:/repository/esb/username"/>

</**pool**>

</**connection**>

</**dblookup**>

| Parameter Name | Description |
| --- | --- |
| **Connection Information** | This parameter is used to specify whether the connection should be taken from a connection pool or a datasource. |
| **Driver** | The class name of the database driver. |
| **URL** | JDBC URL of the database where the data will be looked up. |
| **User** | Username used to connect to the database. |
| **Password** | Password used to connect to the database. |

To add properties to the DBLookup mediator, start with the following parameters:

| Parameter Name | Description |
| --- | --- |
| **Name** | The name of the property. |
| **Value** | The value of the property. |
| **Action** | This parameter enables a property to be deleted. |

Once you have defined the above parameters, enter the following properties:

| Name | Value | Description |
| --- | --- | --- |
| autocommit | true / false | The auto-commit state of the connections created by the pool. |
| isolation | Connection.TRANSACTION\_NONE / Connection.TRANSACTION\_READ\_COMMITTED / Connection.TRANSACTION\_READ\_UNCOMMITTED / Connection.TRANSACTION\_REPEATABLE\_READ / Connection.TRANSACTION\_SERIALIZABLE | The isolation state of the connections created by the pool. |
| initialsize | int | The initial number of connections created when the pool is started. |
| maxactive | int | The maximum number of active connections that can be allocated from this pool at a given time. When this maximum limit is reached, no more active connections will be created by the connection pool. Specify 0 or a negative value if you do not want to set a limit. |
| maxidle | int | The maximum number of idle connections allowed in the connection pool at a given time. The value should be less than the maxActive value. For high performance, tune maxIdle to match the number of average, concurrent requests to the pool. If this value is set to a large value, the pool will contain unnecessary idle connections.  The enabled idle connections are checked periodically whenever a new connection is requested, and connections that are being idle for longer than minEvictableIdleTimeMillis are released, since it takes time to create a new connection. |
| maxopenstatements | int | The maximum number of open statements that can be allocated from the statement pool at a given time. When this maximum limit is reached, no more new statements will be created by the statement pool. Specify 0 or a negative value if you do not want to set a limit. |
| maxwait | long | The maximum number of milliseconds that the connection pool will wait for a connection to return before throwing an exception when there are no connections available in the pool. Specify 0 or a negative value if you want the pool to wait indefinitely. |
| minidle | int | The minimum number of idle connections allowed in the connection pool at a given time, without extra ones being created . Default value is 0, and is derived from initialSize . The connection pool can shrink below this number if validation queries fail.  This value should be similar or near to the average number of requests that will be received by the server at the same time. With this setting, you can avoid having to open and close new connections every time a request is received by the server. |
| poolstatements | true/ false | If the value is true, statement pooling is enabled for the pool. |
| testonborrow | true/ false | If the value is true, objects are validated before they are borrowed from the pool. An object which fails the validation test will be dropped from the pool and another object in the pool will be picked instead. |
| testwhileidle | true/ false | If the value is true , the objects in the pool will be validated using an idle object evictor (if any exists). Any object which fails this validation test would be dropped from the pool. |
| validationquery | String | The SQL query that will be used to validate connections from this pool before returning them to the caller. |

Datasource configurations[¶](https://ei.docs.wso2.com/en/7.0.0/micro-integrator/references/mediators/dBLookup-Mediator/#datasource-configurations)

The configuration of the DBLookup mediator further differs based on whether the connection to the database is made using an external datasource or a Carbon datasource.

The parameters available to configure the DBLookup mediator are as follows.

| Parameter Name | Description |
| --- | --- |
| **Connection Information** | This parameter is used to specify whether the connection should be taken from a connection pool or a datasource. |
| **Datasource Type** | This parameter is used to specify whether the connection to the database should be made using an external datasource or a Carbon datasource. |
| **JNDI Name** | The JNDI used to look up data. |

SQL statements[¶](https://ei.docs.wso2.com/en/7.0.0/micro-integrator/references/mediators/dBLookup-Mediator/#sql-statements)

| Parameter Name | Description |
| --- | --- |
| **SQL** | This parameter is used to enter one or more SQL statements. |
| **Parameters** | This section is used to specify how the values of parameters in the SQL will be determined. A parameter value can be static or calculated at runtime based on a given expression. |
| **Parameter Type** | The data type of the parameter. Possible values are as follows.  **CHAR**  **VARCHAR**  **LONGVARCHAR**  **NUMERIC**  **DECIMAL**  **BIT**  **TINYINT**  **SAMLLINT**  **INTEGER**  **BIGINT**  **REAL**  **DOUBLE**  **DATE**  **TIME**  **TIMESTAMP** |
| **Property Type** | This determines whether the parameter value should be a static value or calculated at run time via an expression.  **Value** : If this is selected, a static value would be considered as the property value and this value should be entered in the **Value/Expression** parameter.  **Expression:** If this is selected, the property value will be determined during mediation by evaluating an expression. This expression should be entered in the **Value/Expression** parameter. |
| **Value/Expression** | This parameter is used to enter the static value or the XPath expression used to determine the property value based on the option you selected for the **Property Type** parameter.  You can click **NameSpaces** to add namespaces if you are providing an expression. Then the **Namespace Editor** panel would appear where you can provide any number of namespace prefixes and URLs used in the XPath expression. |
| **Action** | This allows you to delete a parameter. |
| **Results** | This section is used to specify how to deal with the rerun result from a Database query execution.  **Result Name**  **Column**  **Action** - Deletes the result. |

Example[¶](https://ei.docs.wso2.com/en/7.0.0/micro-integrator/references/mediators/dBLookup-Mediator/#example)

<dblookup xmlns="http://ws.apache.org/ns/synapse">

<connection>

<pool>

<driver>org.apache.derby.jdbc.ClientDriver</driver>

<url>jdbc:derby://localhost:1527/esbdb;create=false</url>

<user>esb</user>

<password>esb</password>

</pool>

</connection>

<statement>

<sql><CDATA[select \* from company where name =?]]></sql>

<parameter expression="//m0:getQuote/m0:request/m0:symbol" type="VARCHAR" xmlns:m0="http://services.samples/xsd"/>

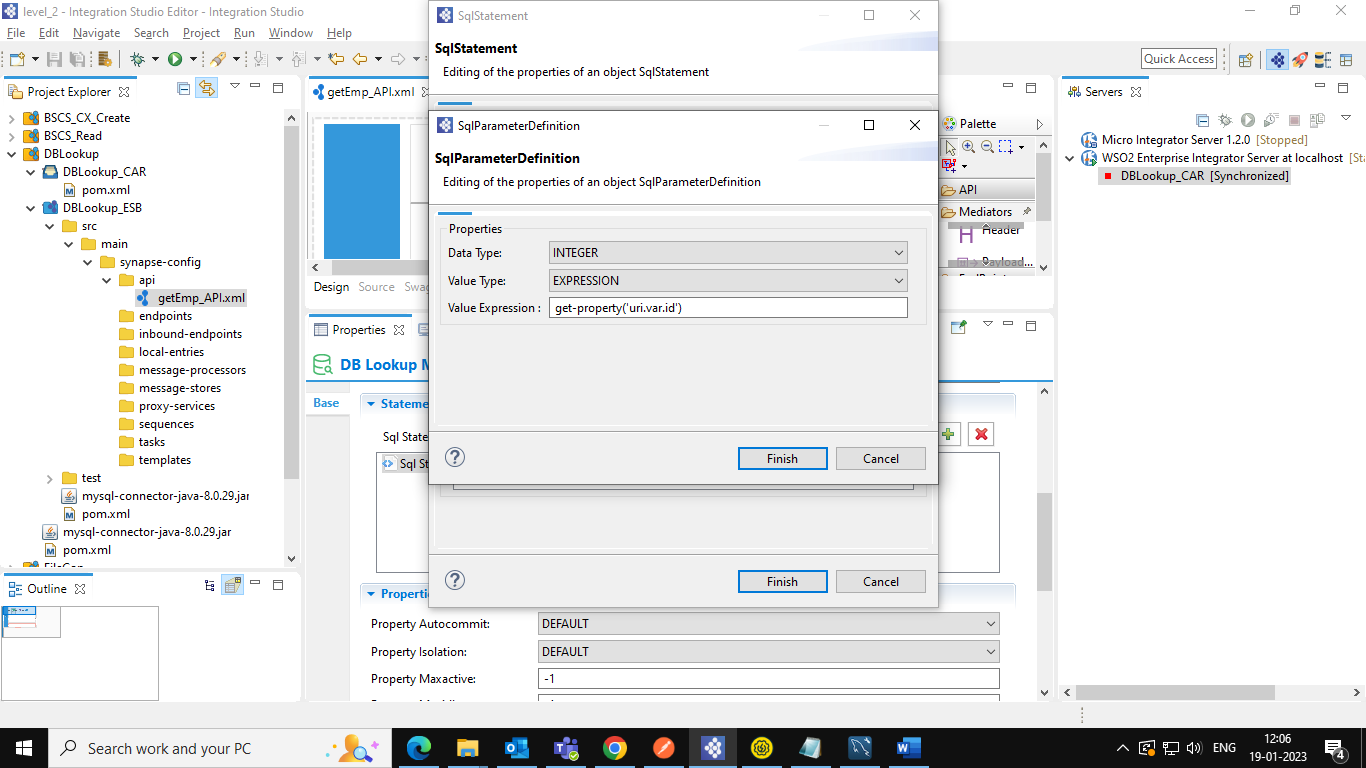
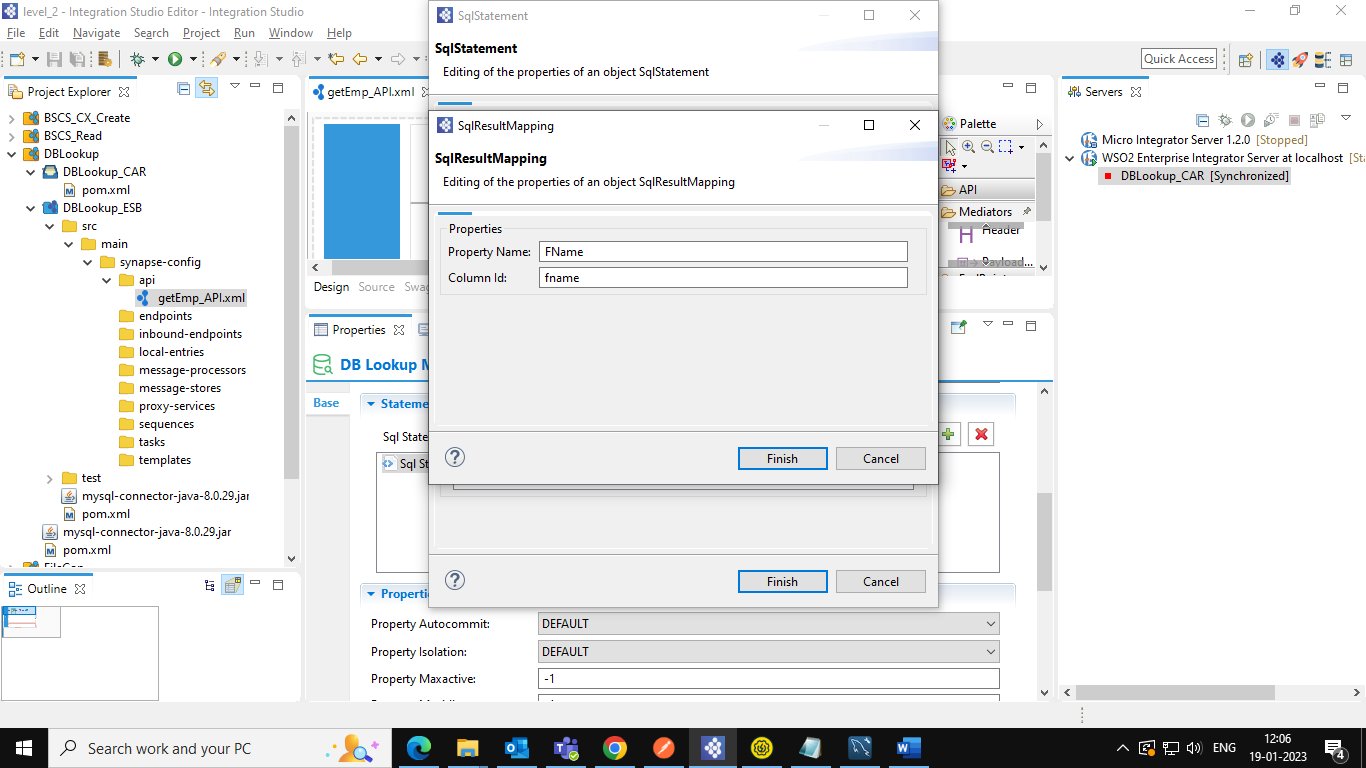
<result column="id" name="company\_id"/>

</statement>

</dblookup>

**MY Snaps:**

![Graphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generatedGraphical user interface, text, application

Description automatically generated