DB Report Mediator[¶](https://ei.docs.wso2.com/en/7.0.0/micro-integrator/references/mediators/dB-Report-Mediator/#db-report-mediator)

The **DB Report Mediator** is similar to the [DBLookup Mediator](https://ei.docs.wso2.com/en/7.0.0/micro-integrator/references/mediators/dBLookup-Mediator/). The difference between the two mediators is that the DB Report mediator writes information to a database using the specified insert SQL statement.

**Info**

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

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

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

**Connection Pool**

<dbreport>

<connection>

<pool>

(

<driver/>

<url/>

<user/>

<password/>

<dsName/>

<icClass/>

<url/>

<user/>

<password/>

)

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

</pool>

</connection>

<statement>

<sql>insert into something values(?, ?, ?, ?)</sql>

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

</statement>+

</dbreport>

**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. Click on the relevant tab to view the required syntax.

External Datasource

<dbreport>

<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"/>\*

</statement>+

</dbreport>

Carbon Datasource

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

The configuration of the DBQuery 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/dB-Report-Mediator/#connection-pool-configurations)

The parameters available to configure the DB Report 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. ```

```

| Parameter Name | Description |
| --- | --- |
| **Use Transaction** | This parameter specifies whether the database operation should be performed within a transaction or not. Click **Yes** or **No** as relevant.  To include multiple database reports within the same database transaction i nside a particular message flow, set the value of this **Use Transaction** property to **Yes** .  However, when you have more reports it takes more time to complete a transaction and when multiple messages flow in, then multiple transactions can become active at the same time.  By default, the maximum number of active transactions is 50 as imposed by the Atomikos JTA implementation. To override this, create a file named transaction.properties by including the following property and add it to the <EI\_HOME>/lib directory:  com.atomikos.icatch.max\_actives=1000  Specifying the value as -1 allows unlimited transactions. Change the value accordingly to limit the number of active transactions based on your environment and the concurrency level of the service.  If you click **Yes** to perform the database operation within a transaction, you need to ensure the following:  The DBReport mediator configuration must be preceded by a Transaction Mediator configuration with new as the transaction action.  The DBReport mediator configuration must be followed by a Transaction Mediator configuration with commit as the transaction action.  For detailed information about configuring Transaction mediators, see [Transaction Mediator](https://ei.docs.wso2.com/en/7.0.0/micro-integrator/references/mediators/transaction-Mediator) . |
| **Driver** | The class name of the database driver. |
| **Url** | The JDBC URL of the database that data will be written to.  Set the autoReconnect parameter to true to help reconnect to the database when the connection between the client and the database is dropped. For example, <url>jdbc:mysql://<ip>:<port>/test?autoReconnect=true</url> . |
| **User** | The user name for connecting to the database. |
| **Password** | The password used to connect to the database. |

To add properties to the DBReport 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 to be allowed in the connection pool at a given time. Specify 0 or a negative value if you want the pool to wait indefinitely. |
| 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 to be allowed in the connection pool at a given time. Specify 0 or a negative value if you want the pool to wait indefinitely. |
| 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.  This property helps to reconnect to the database when the database connection between the client and the database is dropped. For example, <property name="validationquery" value="select 1"/> . |

Datasource configurations[¶](https://ei.docs.wso2.com/en/7.0.0/micro-integrator/references/mediators/dB-Report-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.

**External Datasource**[**¶**](https://ei.docs.wso2.com/en/7.0.0/micro-integrator/references/mediators/dB-Report-Mediator/#external-datasource)

The parameters available to configure the DB Report mediator as an external datasource are as follows.

| Parameter Name | Description |
| --- | --- |
| **Use Transaction** | This parameter specifies whether the database operation should be performed within a transaction or not. Click **Yes** or **No** as relevant. |
| **Initial Context** | The initial context factory class. The corresponding Java environment property is java.naming.factory.initial . |
| **Datasource Name** | The naming service provider URL . The corresponding Java environment property is java.naming.provider.url . |
| **URL** | The JDBC URL of the database that data will be written to. |
| **User** | The user name used to connect to the database. |
| **Password** | The password used to connect to the database. |

To add properties to the DBReport 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 to be allowed in the connection pool at a given time. Specify 0 or a negative value if you want the pool to wait indefinitely. |
| 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 to be allowed in the connection pool at a given time. Specify 0 or a negative value if you want the pool to wait indefinitely. |
| 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. |

**Carbon Datasource**[**¶**](https://ei.docs.wso2.com/en/7.0.0/micro-integrator/references/mediators/dB-Report-Mediator/#carbon-datasource)

| Parameter Name | Description |
| --- | --- |
| **Use Transaction** | This parameter specifies whether the database operation should be performed within a transaction or not. Click **Yes** or **No** as relevant. |
| **Datasource** | This parameter is used to selected a specific Carbon datasource you want to use to make the connection. All the Carbon datasources which are currently available are included in the list. |

SQL statements[¶](https://ei.docs.wso2.com/en/7.0.0/micro-integrator/references/mediators/dB-Report-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. |

Examples[¶](https://ei.docs.wso2.com/en/7.0.0/micro-integrator/references/mediators/dB-Report-Mediator/#examples)

Simple database write operation[¶](https://ei.docs.wso2.com/en/7.0.0/micro-integrator/references/mediators/dB-Report-Mediator/#simple-database-write-operation)

This example demonstrates simple database write operations. The DB Report mediator writes to a table using the details of the message. It updates the stock price of the company using the last quote value, which is calculated by evaluating an XPath expression against the response message.

<dbreport 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[update company set price=? where name =?]]></sql>

<parameter expression="//m0:return/m1:last/child::text()" type="DOUBLE" xmlns:m0="http://services.samples" xmlns:m1="http://services.samples/xsd"/>

<parameter expression="//m0:return/m1:symbol/child::text()" type="VARCHAR" xmlns:m0="http://services.samples" xmlns:m1="http://services.samples/xsd"/>

</statement>

</dbreport>

Database write operation within a transaction[¶](https://ei.docs.wso2.com/en/7.0.0/micro-integrator/references/mediators/dB-Report-Mediator/#database-write-operation-within-a-transaction)

In this example, <transaction action="new"/> is a Transaction Mediator configuration that starts a new transaction. The DBReport mediator configuration performs a few write operations incuding deleting records when the name matches a specific value derived via an expression as well as a few insertions. Once the database operations are complete, they are committed via <transaction action="commit"/> , which is another Transaction Mediator configuration.

<sequence xmlns="http://ws.apache.org/ns/synapse" name="myFaultHandler">

<log level="custom">

<property name="text" value="\*\* Rollback Transaction\*\*"/>

</log>

<transaction action="rollback"/>

<send/>

</sequence>

<proxy name="SimpleProxy" transports="http https" startonload="true" trace="disable" xmlns="http://ws.apache.org/ns/synapse">

<target>

<inSequence>

<send>

<endpoint>

<address uri="http://localhost:9000/services/SimpleStockQuoteService"/>

</endpoint>

</send>

</inSequence>

<outSequence>

<transaction action="new"/>

<log level="custom">

<property name="text" value="\*\* Reporting to the Database EIdb\*\*"/>

</log>

<dbreport useTransaction="true" xmlns="http://ws.apache.org/ns/synapse">

<connection>

<pool>

<dsName>java:jdbc/XADerbyDS</dsName>

<icClass>org.jnp.interfaces.NamingContextFactory</icClass>

<url>localhost:1099</url>

<user>EI</user>

<password>EI</password>

</pool>

</connection>

<statement>

<sql>delete from company where name =?</sql>

<parameter expression="//m0:return/m1:symbol/child::text()"

xmlns:m0="http://services.samples" xmlns:m1="http://services.samples/xsd"

type="VARCHAR"/>

</statement>

</dbreport>

<log level="custom">

<property name="text" value="\*\* Reporting to the Database EIdb1\*\*"/>

</log>

<dbreport useTransaction="true" xmlns="http://ws.apache.org/ns/synapse">

<connection>

<pool>

<dsName>java:jdbc/XADerbyDS1</dsName>

<icClass>org.jnp.interfaces.NamingContextFactory</icClass>

<url>localhost:1099</url>

<user>EI</user>

<password>EI</password>

</pool>

</connection>

<statement>

<sql>INSERT into company values (?,'c4',?)</sql>

<parameter expression="//m0:return/m1:symbol/child::text()"

xmlns:m1="http://services.samples/xsd" xmlns:m0="http://services.samples"

type="VARCHAR"/>

<parameter expression="//m0:return/m1:last/child::text()"

xmlns:m1="http://services.samples/xsd" xmlns:m0="http://services.samples"

type="DOUBLE"/>

</statement>

</dbreport>

<transaction action="commit"/>

<send/>

</outSequence>

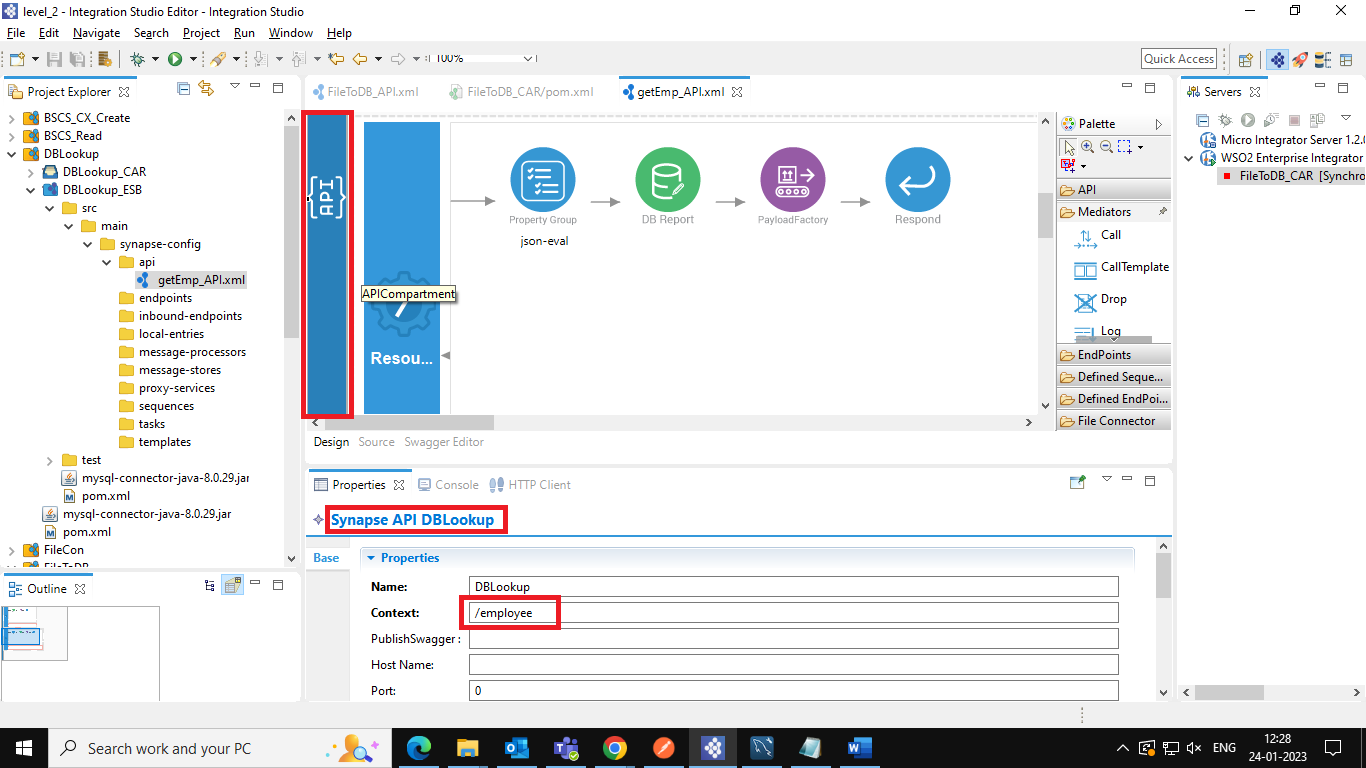
<faultSequence>

<sequence key="myFaultHandler"/>

</faultSequence>

</target>

</proxy>



Graphical user interface, text, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

A screenshot of a computer

Description automatically generated

