Database Connector: Manual for Version 1.1.0

Alexander Seitinger
Software Engineer

Medexter Healthcare GmbH
Borschkegasse 7/5, A-1090 Vienna, Austria

+43-1-968 03 24 tel
+43-1-968 09 22 fax
as@medexter.com
www.medexter.com

2014-11-20

Table of Contents

1. Introduction

This document describes the database connector module, an optional component of the Arden Syntax server. This component allows the Arden Syntax server to access an external database from within medical logic modules (MLMs) by using so-called curly brace expressions. Whenever the Arden Syntax execution engine, which is part of the Arden Syntax server back end, has to process a curly brace expression, the expression is sent to the database connector module. A database connection is provided and the curly brace expression is sent to the corresponding database as a query. Retrieved data elements are transformed into Arden Syntax data types and returned to the engine that made the request. Since the Arden Syntax specification does not regulate the content of curly brace expressions and their validation is declared institution-specific, there are no restrictions regarding their content. It is the MLM author's responsibility to provide valid expressions. The database connector does not verify validity. If an error occurs during execution of a statement, the corresponding error message is returned as a string. Any database vendor that provides a JDBC driver may be used with this module.

The database connector's web interface allows for management of multiple database connections. The user may create, edit and delete connections. It is also possible to specify which of the existing connections should be used by the database connector module to execute incoming curly brace expressions. Only one connection can be active at a time.

2. Quick Installation Guide

In order to install the database connector module properly, please ensure that the following steps are carried out (assuming that an Arden Syntax server is already installed and running):

1. Copy the respective JDBC 4.0 driver you wish to use to the following directory (see paragraph 3.2 for further details):

   ```
   $SERVER_PATH/glassfish/domains/tamAS/lib/ext
   ```

2. Open a browser window and access the glassfish admin console by entering this URL into the address bar:

   ```
   http://localhost:4848/common/index.jsf
   ```

3. If you haven't been assigned any other username and password, use *admin* (username) and *adminadmin* (password) to login.

4. Go to *Common Tasks → Applications*, then press the 'Deploy' button and select your **dbConnectorModule.war** file as *packaged file to be uploaded to the server*. Press OK.

5. Deploy the **dbConnectorFrontend.war** file in the same fashion.

6. Check if the **dbConnector_custom.properties** file contains any property that has to be changed (see paragraph 3.1 for further details):

   ```
   $SERVER_PATH\glassfish\domains\tamAS\config\dbConnector_custom.properties
   ```

7. Restart your Arden Syntax server for all changes to take effect (JDBC 4.0 file, properties).

3. Administrator Guide

The database connector module is intended to be used as an add-on to the Arden Syntax server. For it to work, the module's front end has to be deployed on the same glassfish server as the Arden Syntax server front end, since these two components share the same authentication mechanism to manage access rights.

3.1. Setup

The database connector module consists of two components:

dbConnectorModule.war
dbConnectorFrontend.war

Utilizing the Java naming and directory interface (JNDI), it is possible to deploy these files onto different servers. Two property files have to be adapted accordingly:

```
$SERVER_PATH\glassfish\domains\tamAS\config\ardenServer.properties
$SERVER_PATH\glassfish\domains\tamAS\config\dbConnector_custom.properties
```

For a description of the **ardenServer.properties** file, please check the Arden Syntax server handbook.

The database connector property file **dbConnector_custom.properties** contains the following sections:

3.1.1. Remote Connection Properties

The remote connections section specifies the listening port which is used by the module to receive RMI calls as well as the address of the server hosting it.

```
RmiServicePort=19898
RmiServiceHost=127.0.0.1
```

Note: When the database connector module is deployed on a machine with the IP 192.168.0.101, this IP has to be specified as RmiServiceHost. The same IP and port have to be specified in the Arden Syntax server's **ardenServer.properties** file, in order to be able to reach the connector module.

3.1.2. Communication Properties

The communication section contains all settings for the internal communication between the database connector's back end and front end. These settings may be used if components are distributed onto different servers.

```
java.naming.factory.initial=com.sun.enterprise.naming.SerialInitContextFactory
java.naming.factory.url.pkgs=com.sun.enterprise.naming
java.naming.factory.state=com.sun.corba.ee.impl.presentation.rmi.JNDIStateFact
oryImpl
org.omg.CORBA.ORBInitialHost=127.0.0.1
org.omg.CORBA.ORBInitialPort=3700
```

When deploying the database connector's back end, glassfish automatically generates a specified database schema in order to store database connections specified by the user.

Note: The ORBInitialHost address is used by all other database connector components to look up the connector's back end via JNDI and therefore has to equal the IP of the machine on which this back end component is deployed. The ORBInitialPort equals the IIOP port specified during the installation of the glassfish server.

As long as the database connector module is deployed on the same glassfish server as the Arden Syntax server back end, all necessary settings are already preset. If deployed otherwise, please check the Arden Syntax server manual for further information (JDBC Resources, JDBC Connection Pools).

3.2. Providing Other JDBC Drivers

JDBC drivers are used by the database connector module to connect to different database vendors (e.g., MySQL, Oracle, Microsoft SQL). In principle, the database connector module can be used with any JDBC driver. However, only JDBC 4.0 drivers provide automatic driver loading, using the Java SE service provider mechanism. In order to use the module with different JDBC 4.0 drivers, the responsible server administrator simply has to copy the respective **JDBC.jar** file into the corresponding domain's lib folder and restart the server:

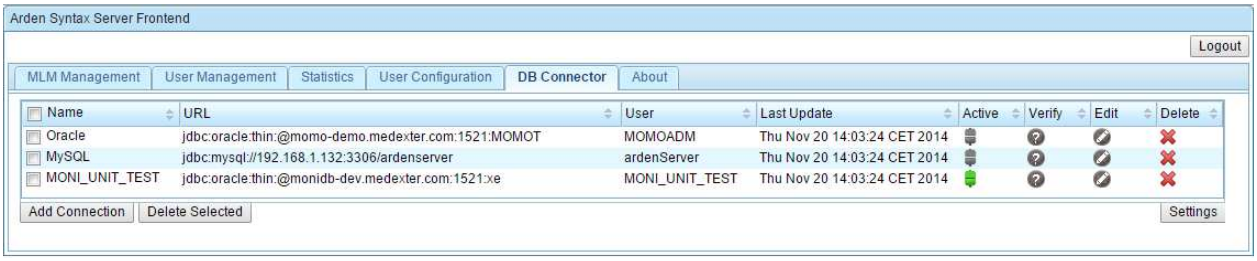
```
$SERVER_PATH/glassfish/domains/tamAS/lib/ext
```

If you want to use an older driver version, please contact our support team at [Medexter Healthcare](). To check if a driver supports JDBC 4.0, please open its .jar file with a decompression program like 7zip or Winrar and see if it contains a META-INF/services/java.sql.Driver file.

4. User Guide

After successfully configuring and starting the database connector module, its web front end is available:

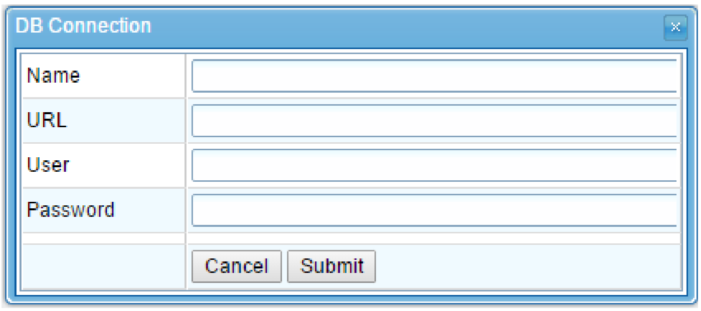
```
http://$SERVERNAME:8080/ArdenSyntaxServerFrontend/start.zul#tab5
```



□ Name	URL	User	Last Update	Active	Verify	Edit	Delete
□ Oracle	jdbc:oracle:thin:@momo-demo.medexter.com:1521:MOMOT	MOMOADM	Thu Nov 20 14:03:24 CET 2014	●	?	✎	✗
□ MySQL	jdbc:mysql://192.168.1.132:3306/ardenserver	ardenServer	Thu Nov 20 14:03:24 CET 2014	●	?	✎	✗
□ MONI_UNIT_TEST	jdbc:oracle:thin:@monidb-dev.medexter.com:1521:xe	MONI_UNIT_TEST	Thu Nov 20 14:03:24 CET 2014	🟩	?	✎	✗

Figure 1: The database connector's user interface

The user interface provides the means to add, verify, edit and delete connections. A connection can be added by clicking the 'Add Connection' button. This opens up a form in which the user has to specify a JDBC connection URL as well as his username and password. The user must have permission to establish a connection to the selected database. For your convenience it is possible to assign a name to each connection so you can quickly identify them.



DB Connection

Name	
URL	
User	
Password	
	Cancel Submit

Figure 2: Form for a new connection entry

Any database vendor providing JDBC drivers may be used. Each vendor uses a unique connection string (URL), e.g.:

`IBM DB2:`	`jdbc:db2://<HOST>:<PORT>/<DB>`
`ODBC Bridge:`	`jdbc:odbc:<DB>`
`Microsoft SQL Server:`	`jdbc:weblogic:mssqlserver4:<DB>@<HOST>:<PORT>`
`Oracle Thin:`	`jdbc:oracle:thin:@<HOST>:<PORT>:<SID>`
`MySQL:`	`jdbc:mysql://<HOST>:<PORT>/<DB>`
`PostgreSQL:`	`jdbc:postgresql://<HOST>:<PORT>/<DB>`

The database connection module was tested using JDBC drivers for a MySQL and Oracle database. Connection string example:

```
jdbc:mysql://192.168.1.144:3306/ardendatabase
```

When pressing 'Submit', provided information is stored. For each stored connection, four actions may be performed:

4.1. User Actions

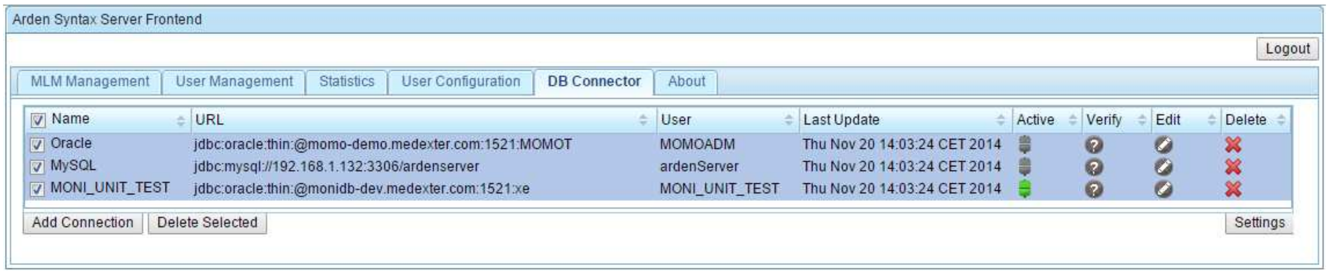
🟩/⬛ Set selected connection active/inactive: As soon as the connection is active, curly brace expressions inside MLMs will be executed as SQL statements using this connection.

? Verify connection: Checks if a connection can be established using provided configuration. If a connection could be established, the symbol changes to: ✓ Otherwise, ✖ is displayed.

✎ Edit connection name, URL, username and password. A connection has to be inactive in order to be able to edit its connection data.

✖ Delete selected connection.

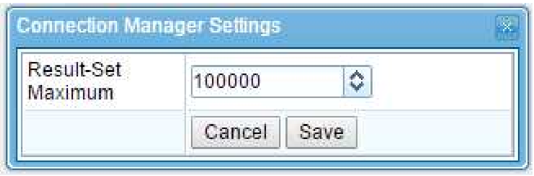
To delete one or multiple connections, the user can also use the checkboxes on the far left side of each connection entry and press 'Delete selected'.



☑ Name	URL	User	Last Update	Active	Verify	Edit	Delete
☑ Oracle	jdbc:oracle:thin:@momo-demo.medexter.com:1521:MOMOT	MOMOADM	Thu Nov 20 14:03:24 CET 2014	○	?	✎	✗
☑ MySQL	jdbc:mysql://192.168.1.132:3306/ardenserver	ardenServer	Thu Nov 20 14:03:24 CET 2014	○	?	✎	✗
☑ MONI_UNIT_TEST	jdbc:oracle:thin:@monidb-dev.medexter.com:1521:xe	MONI_UNIT_TEST	Thu Nov 20 14:03:24 CET 2014	●	?	✎	✗

Figure 3: Selecting multiple entries

The button 'Settings' on the right side of figure 3 opens a window where the user can specify a result set limit. This limit determines the maximum number of rows or data sets that the module is actually allowed to retrieve from the database, transform into Arden Syntax data types and return to MLMs requesting this data:



Result-Set Maximum: 100000

Figure 4: Customizing settings

4.2. Data Type Conversion

As already mentioned in the introduction, data retrieved using curly brace expressions is transformed into Arden Syntax data types by the database connector module before being returned to the requesting Arden Syntax execution engine. As a result, either a single Arden Syntax data type, a flat list, a two dimensional list, or an Arden object is returned. The conversion is illustrated below:

ResultSet returned by the SQL query:	**Converted to:**
one row, one column	Instance of an Arden Syntax data type
multiple rows, one column	List containing Arden Syntax data type instances as elements
one row, multiple columns	List containing Arden Syntax data type instances as elements
multiple rows, multiple columns	Two dimensional list: Each row becomes an element of a list; each of these elements contains the Arden Syntax data types the row consisted of, also as a list.

For the database connector module to be able to return an ArdenObject (Arden Syntax: read as), each field name of a given object declaration has to occur as column in the result set returned by the SQL query.

As an additional feature, an MLM author can assign “PrimaryTime” (case insensitive) as a temporary column name for one of the column names in a query, e.g.,

```
SELECT “name”, “date” as “PrimaryTime” FROM test_schema;
```

The database connector module will use the data of this column to assign it as primary time when converting the remaining row columns into Arden Syntax data types. If no primary time column is specified, those primary time components are set to null.