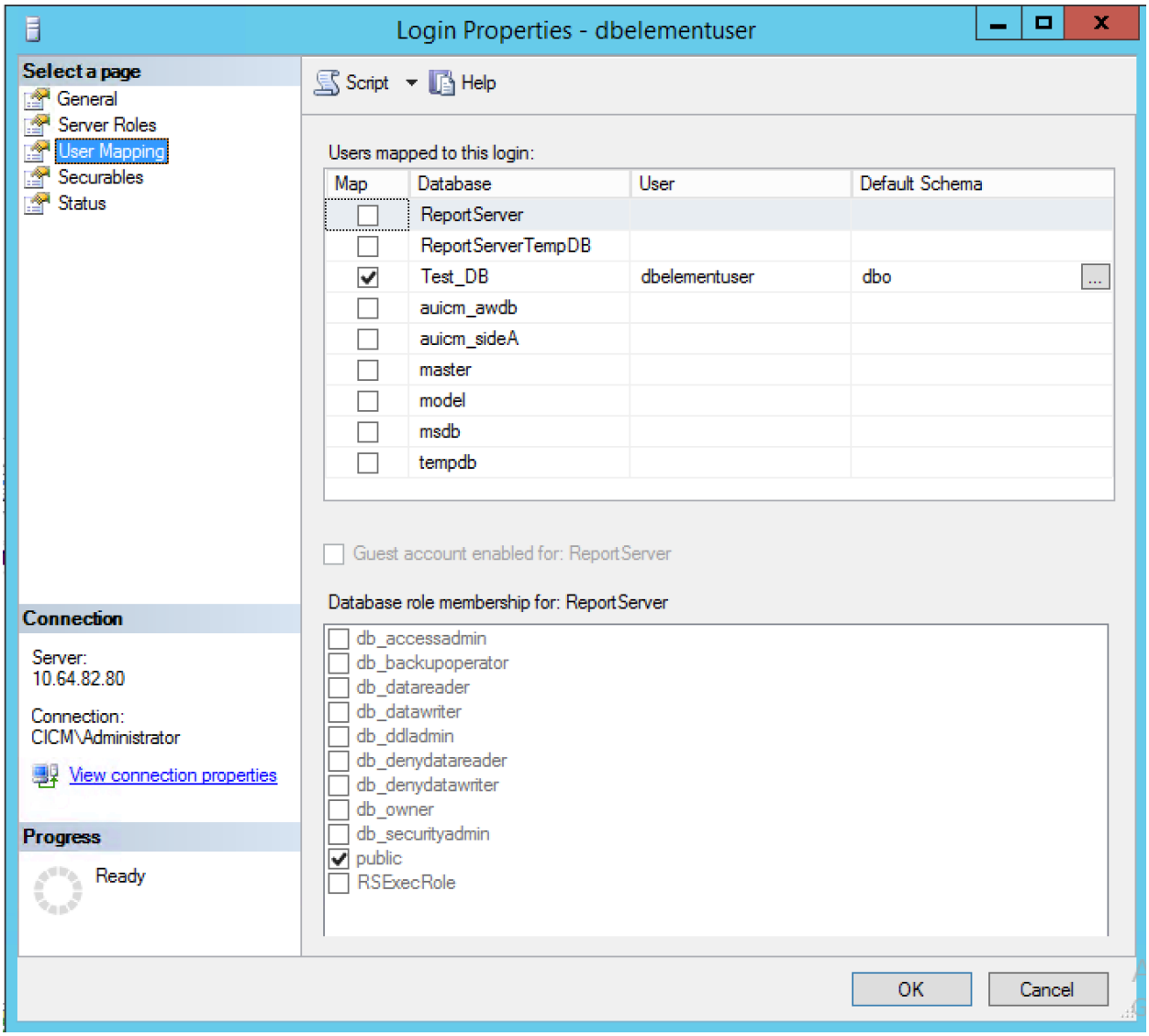
Configuration of Database Action Element in CVP Call Studio

These documented steps are tested with CVP 11.6(1) version, Microsoft SQL Server 2014 SP2 and Windows 2012 R2 OS

External Database Configuration Example

- Create a database as "Test_DB" in SQL Server
- Create a table under this DB as "Test_Table" with the column "Name" and "Grade"
- Add a new SQL user with required permission under Security Logins [Typically this user will have read and write access]. Here the new user is "dblemenetuser"

Login Config example:



CVP VXML Server Configuration

Copying JDBC Driver to VXML server

In order enable database access on your application server, a compatible JDBC driver must be installed. These drivers, typically packaged as JAR files, should be placed in a directory accessible to the application server classpath (on Tomcat, for example, place in %CVP_HOME%\VXMLServer\Tomcat\lib).

Server.xml configuration on VXML server

Add a Tomcat Context for the database connection so that the CVP VXML Server knows how to communicate with your database (C:\Cisco\CVP\VXMLServer\Tomcat\conf\server.xml)

```
<Context>
<Resource name="jdbc/SQLConn"
auth="Container"
type="javax.sql.DataSource"
driverClassName="com.microsoft.sqlserver.jdbc.SQLServerDriver"
url="jdbc:sqlserver://<IP Address of external
Database>:1433;database=Test_DB;user=dbelementuser;password=<password>" />
</Context>
```

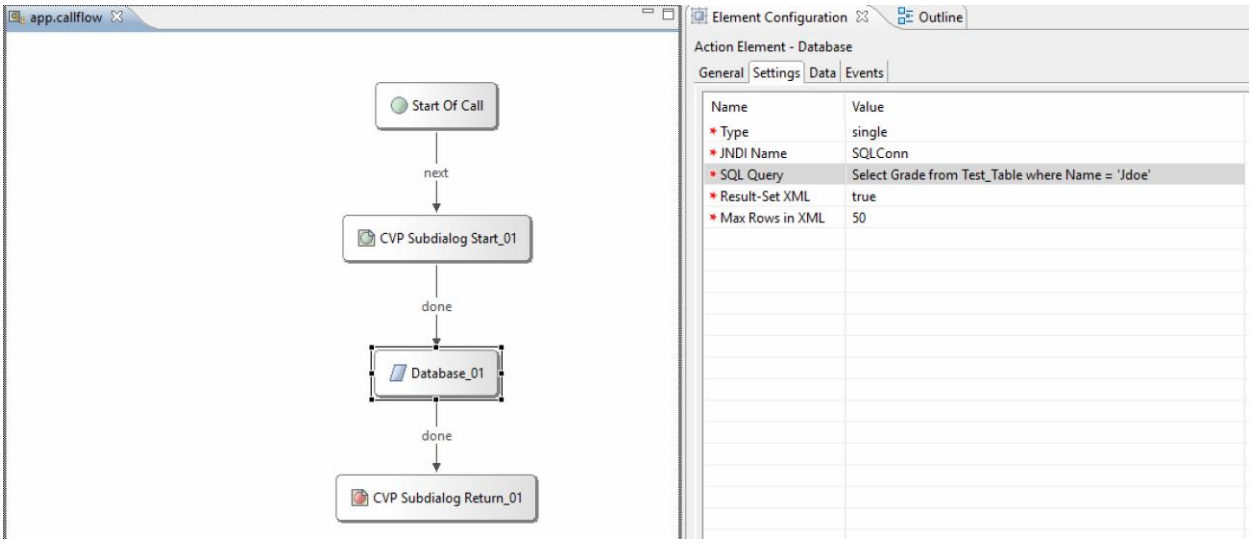
Note:

- Restart the VXML server service once the server.xml configuration/updating the JDBC driver are completed.
- Enter the string as "SQLConn" from the Tomcat Context into the JNDI Name property of the Settings tab of your Database element.

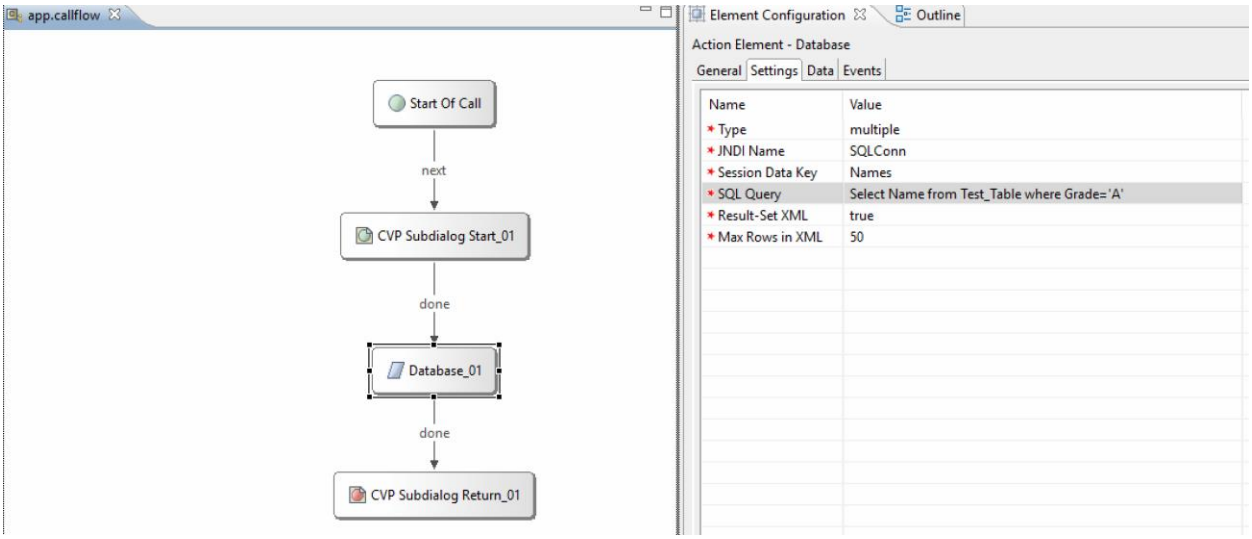
Deploy the application: Exported call studio application is attached

Screenshot of the configuration is as below

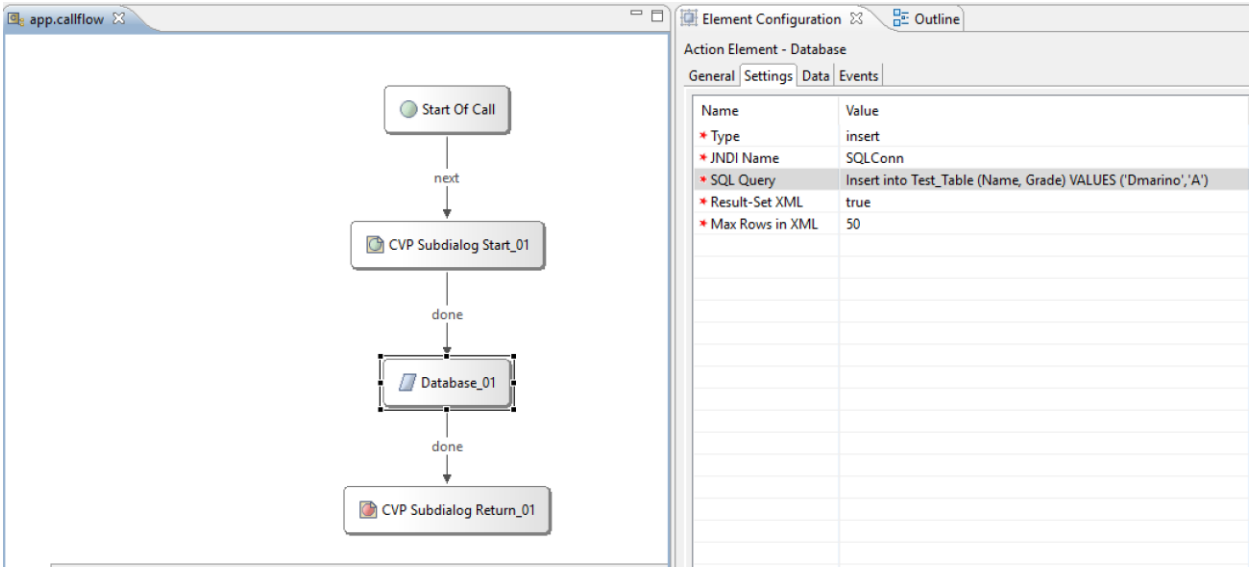
Select Single Row



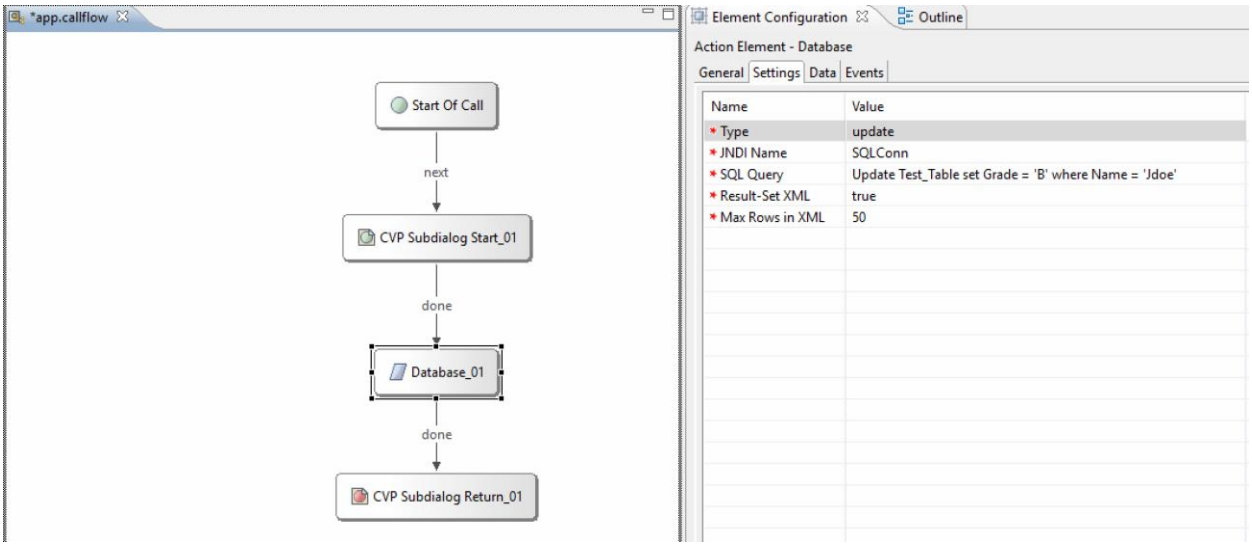
Select Multiple Row



Insert statement



Update statement



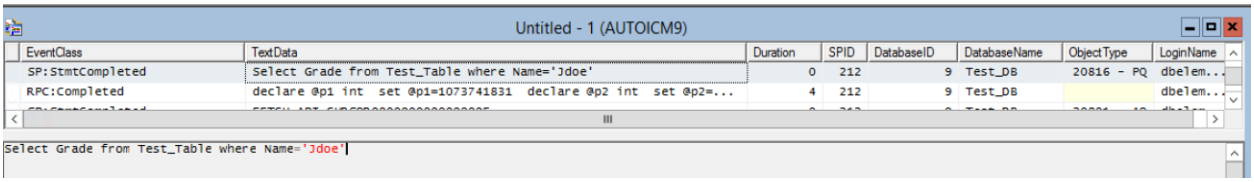
Verification:

Select Single Row

Activity Logs:

10.64.82.142.1530437850360.3.Database_Element,07/01/2018 15:07:30.376,Database_01,enter,
10.64.82.142.1530437850360.3.Database_Element,07/01/2018 15:07:30.563,Database_01,custom,num_rows_processed,1
10.64.82.142.1530437850360.3.Database_Element,07/01/2018 15:07:30.579,Database_01,exit,done

SQL Profiler Logs:

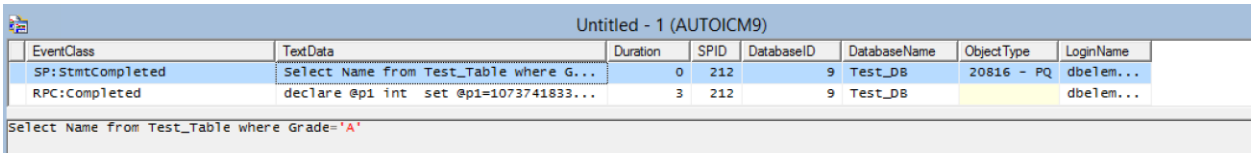


Select Multiple Row

Activity Logs:

10.64.82.142.1530439046253.18.Database_Element,07/01/2018 15:27:26.285,Database_01,enter,
10.64.82.142.1530439046253.18.Database_Element,07/01/2018 15:27:26.316,Database_01,custom,num_rows_processed,2
10.64.82.142.1530439046253.18.Database_Element,07/01/2018 15:27:26.316,Database_01,exit,done

SQL Profiler Logs:

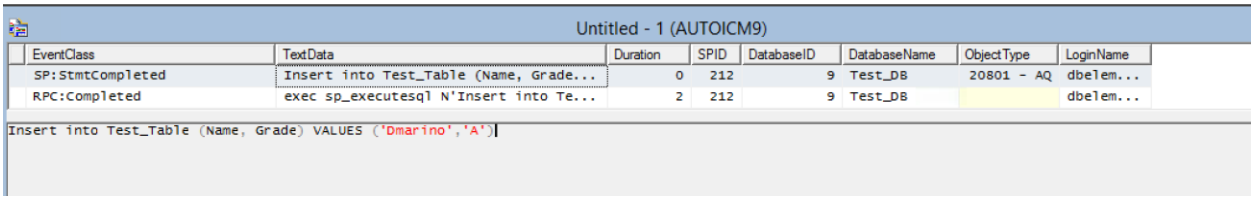


Insert statement

Activity Logs:

10.64.82.142.1530438854878.16.Database_Element,07/01/2018 15:24:14.894,Database_01,enter,
10.64.82.142.1530438854878.16.Database_Element,07/01/2018 15:24:14.909,Database_01,custom,num_rows_processed,1
10.64.82.142.1530438854878.16.Database_Element,07/01/2018 15:24:14.909,Database_01,exit,done

SQL Profiler Logs:



Update statement

Activity Logs:

1 0 .64.82.142.1530439578426.30.Database_Element,07/01/2018 15:36:18.442,Database_01,enter,
1 0 .64.82.142.1530439578426.30.Database_Element,07/01/2018 15:36:18.458,Database_01,custom,num_rows_processed,1
1 0 .64.82.142.1530439578426.30.Database_Element,07/01/2018 15:36:18.458,Database_01,exit,done

SQL Profiler Logs: