iWay Software

How-to's

Creating, Testing, and Deploying a Data Quality Online Service

This topic describes how to create a Data Quality (DQ) online service, test, and then deploy it as a runtime service.

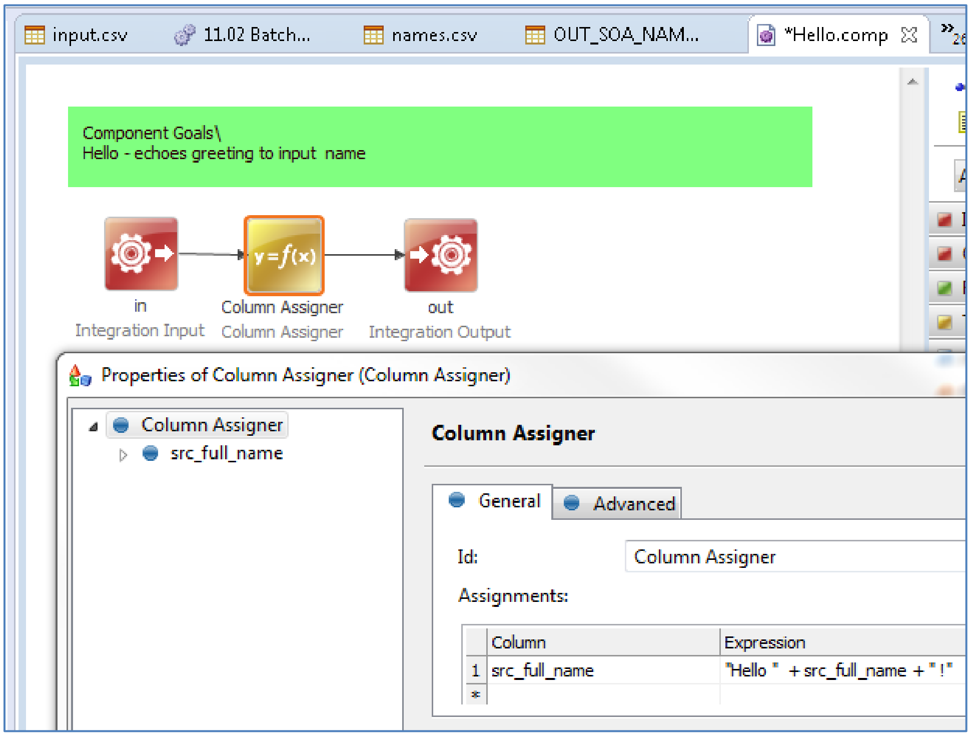
This how-to includes the following components that are packaged in the *samples_dqs.zip* archive:

- **Hello.comp** – Sample DQ component file.
- **Hello.online** – Sample DQ online (web) service file.
- **IN_HELLO.csv** – Sample input CSV file.

This how-to includes the following topics:

Configuring a Simple Component

A Data Quality (DQ) online service must be based on a DQ component (.comp file) with an integration input and output.



Component Goals\
Hello - echoes greeting to input name

Id: Column Assigner

Assignments:

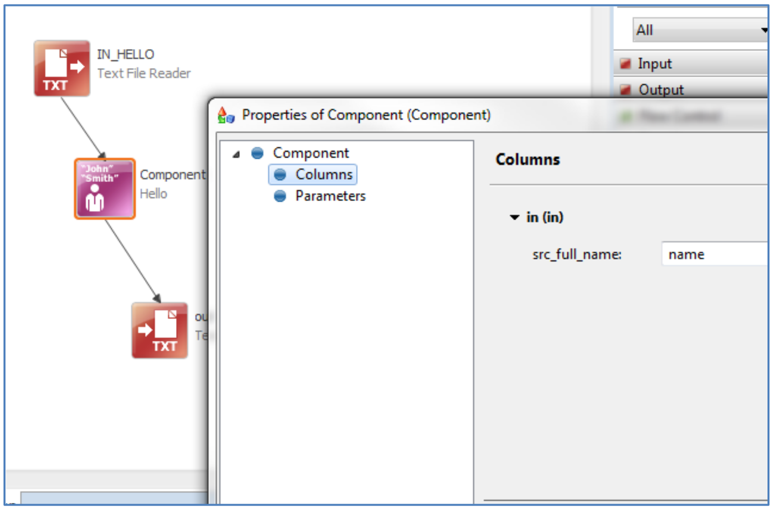
	Column	Expression
1	src_full_name	"Hello " + src_full_name + "!"
*		

The sample component that is included with this how-to (*Hello.comp*) accepts a single field (*src_full_name*) as the input and adds a greeting, which is returned.

Configuring a Test Plan

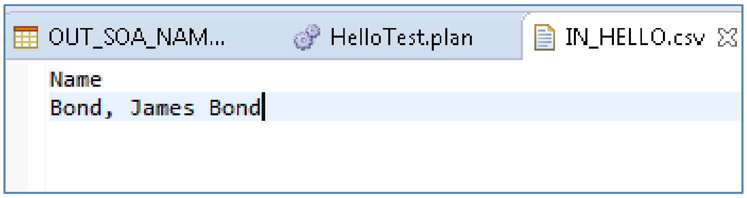
You can test the component locally by calling it from a test plan.

1. Drag and drop the component (*Hello.comp*) onto a test plan.



src_full_name: name

2. Create a simple input CSV file containing a column called *Name* and drag it onto the plan.



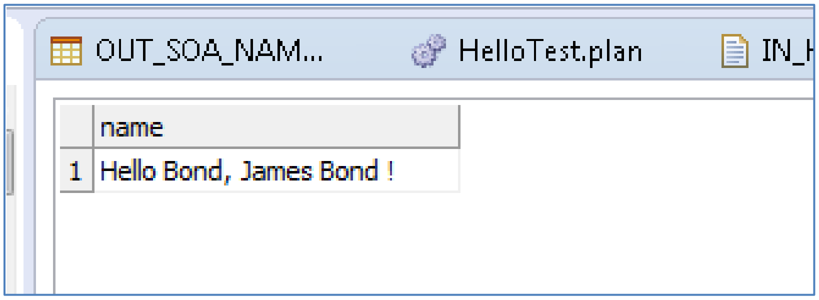
```
Name
Bond, James Bond
```

3. Map the input column from the input CSV file (for example, *Name*) to the column used by the *Hello.comp* component (*src_full_name*).

4. Add a Text File Writer step to the plan and connect all of the components.

5. Edit and save the component logic. Test the plan locally until you are satisfied with the result.

   The following image shows the sample output that is returned by the plan.

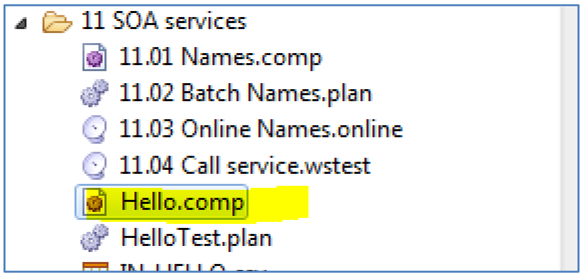


	name
1	Hello Bond, James Bond !

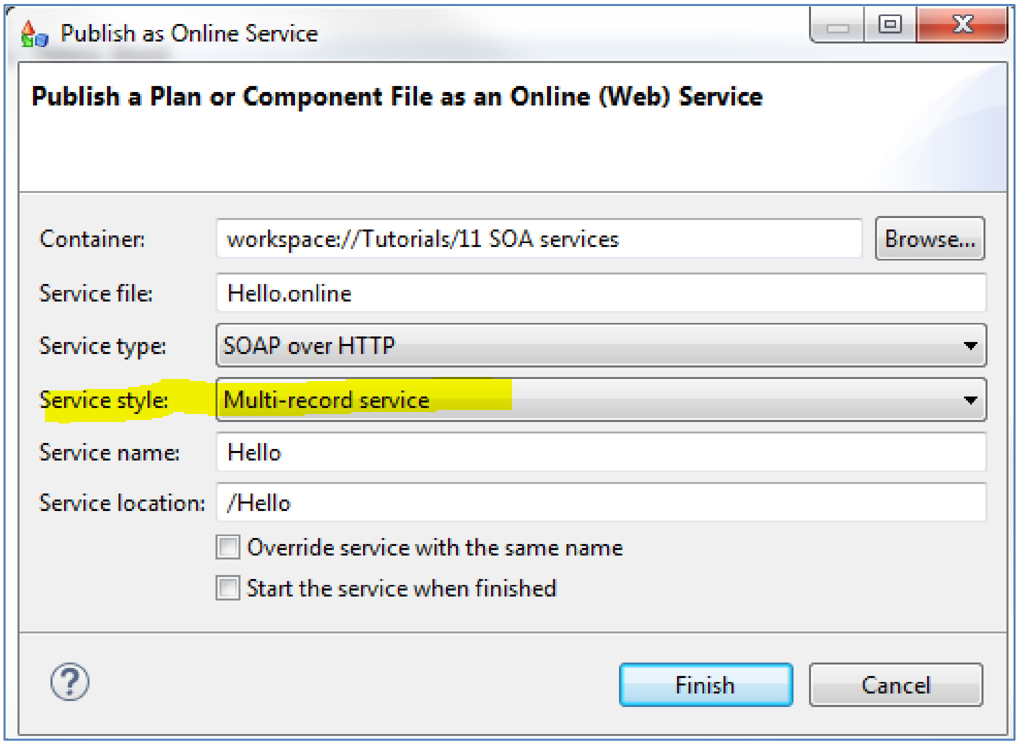
Configuring a Web Service

To configure a web service:

1. Right-click your .comp file (for example, *Hello.comp*) and select *Publish as Online Service* from the context menu.



   The Publish as Online Service dialog opens, as shown in the following image.



Publish as Online Service

Publish a Plan or Component File as an Online (Web) Service

Container: workspace://Tutorials/11 SOA services

Service file: Hello.online

Service type: SOAP over HTTP

Service style: Multi-record service

Service name: Hello

Service location: /Hello

☐ Override service with the same name

☐ Start the service when finished

2. From the Service style drop-down list, select *Multi-record service*.

3. Click *Finish*.

   A corresponding .online file is created (for example, *Hello.online*).

   Note: If you modify the input or output fields in the component at any point, then the .online file must be republished.

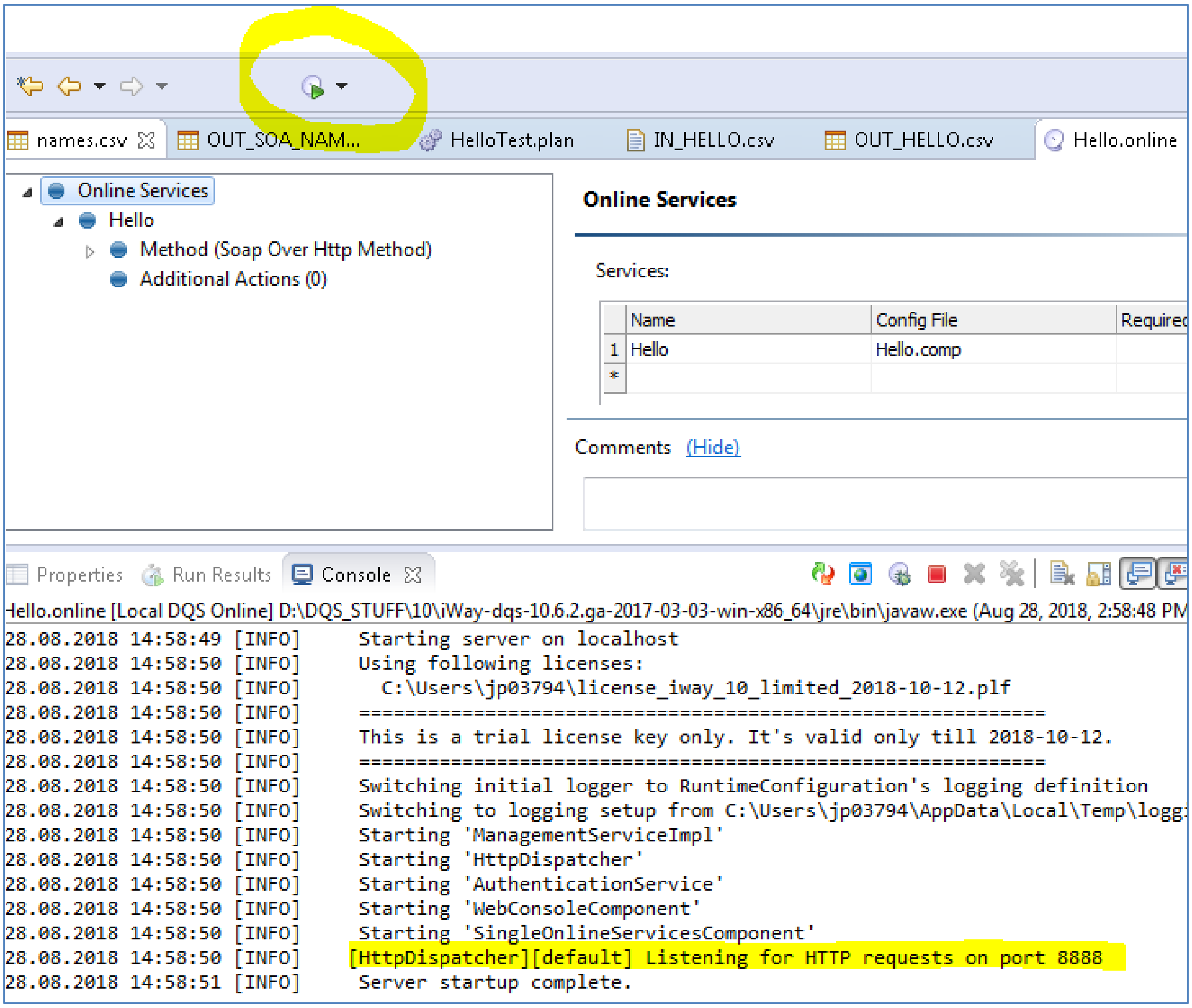
Starting the Online Service in the DQ GUI

You must first start the online server prior to starting the online services. You can start the server from the GUI, but it is not recommended since it is not intended to be used in a production environment.

Only one service can be started at a time. It is also not possible to set a custom password for restricted services requiring a user role. Starting the server directly from the DQ GUI should only be done for debugging purposes.

To start the online service in the DQ GUI:

1. Double-click the .online file (for example, *Hello.online*), which enables the Run button, as shown in the following image.



Online Services

Services:

	Name	Config File	Required
1	Hello	Hello.comp	
*			

Comments (Hide)

```
Hello.online [Local DQS Online] D:\DQS_STUFF\10\iWay-dqs-10.6.2.ga-2017-03-03-win-x86_64\jre\bin\javaw.exe (Aug 28, 2018, 2:58:48 PM
28.08.2018 14:58:49 [INFO]     Starting server on localhost
28.08.2018 14:58:50 [INFO]     Using following licenses:
28.08.2018 14:58:50 [INFO]       C:\Users\jp03794\license_iway_10_limited_2018-10-12.plf
28.08.2018 14:58:50 [INFO]     ============================================================
28.08.2018 14:58:50 [INFO]     This is a trial license key only. It's valid only till 2018-10-12.
28.08.2018 14:58:50 [INFO]     ============================================================
28.08.2018 14:58:50 [INFO]     Switching initial logger to RuntimeConfiguration's logging definition
28.08.2018 14:58:50 [INFO]     Switching to logging setup from C:\Users\jp03794\AppData\Local\Temp\logg
28.08.2018 14:58:50 [INFO]     Starting 'ManagementServiceImpl'
28.08.2018 14:58:50 [INFO]     Starting 'HttpDispatcher'
28.08.2018 14:58:50 [INFO]     Starting 'AuthenticationService'
28.08.2018 14:58:50 [INFO]     Starting 'WebConsoleComponent'
28.08.2018 14:58:50 [INFO]     Starting 'SingleOnlineServicesComponent'
28.08.2018 14:58:50 [INFO]    [HttpDispatcher][default] Listening for HTTP requests on port 8888
28.08.2018 14:58:51 [INFO]     Server startup complete.
```

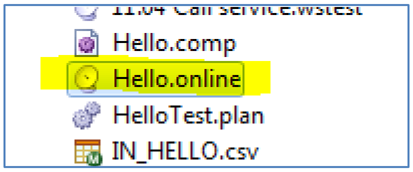
2. Click the *Run* button.

   Note that this service is listening on port 8888.

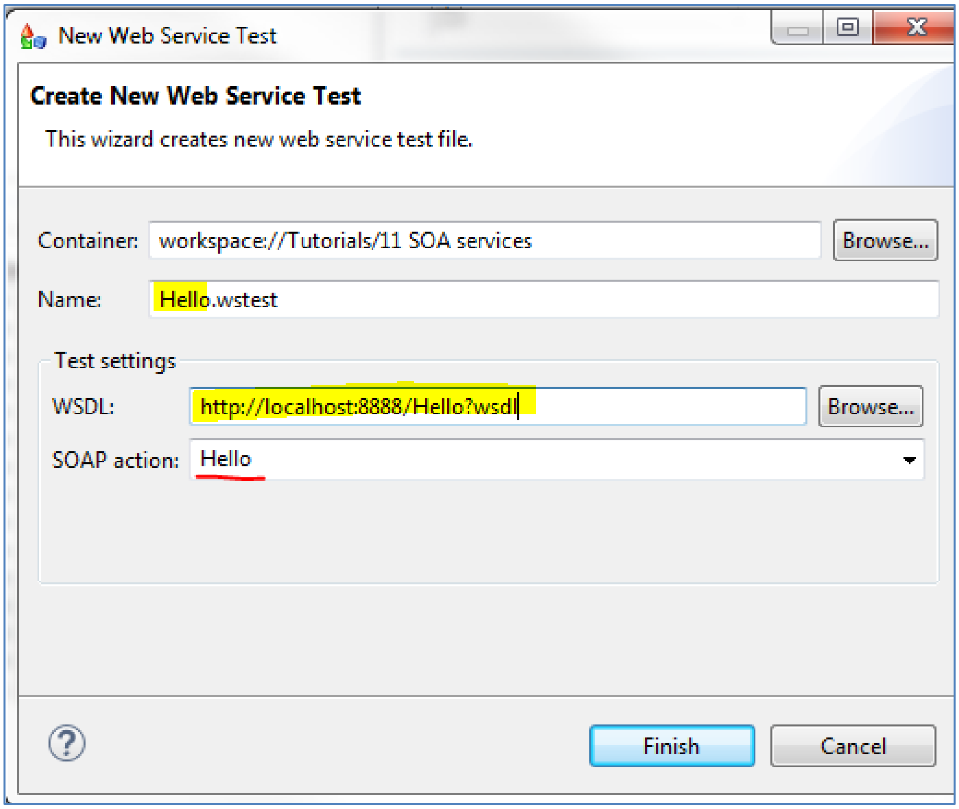
Creating a Web Service Test File

Although you can use any web service client tool to test the online service, the DQ GUI provides a web service utility for your convenience.

1. Right-click the .online plan (for example, *Hello.online*), click *New* from the context menu, and then select *Web Service Test*.



   The New Web Service Test dialog opens, as shown in the following image.



New Web Service Test

Create New Web Service Test

This wizard creates new web service test file.

Container: workspace://Tutorials/11 SOA services

Name: Hello.wstest

Test settings

WSDL: http://localhost:8888/Hello?wsdl

SOAP action: Hello

2. Enter a name and a WSDL URL in the corresponding fields.

   Specify the WSDL URL using the following format:

   `http://localhost:8888/`***`myservicename`***`?wsdl`

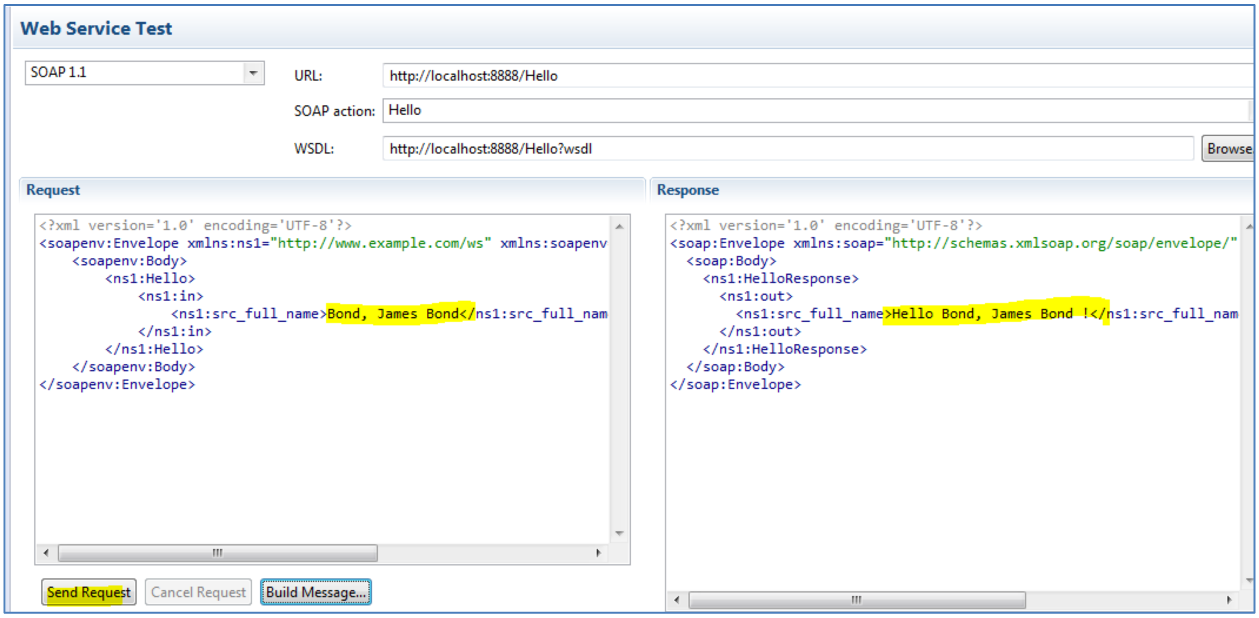
   For example:

   `http://localhost:8888/`**`Hello`**`?wsdl`

   The SOAP action field will be automatically populated provided the WSDL URL is valid and the service was started.

3. Click *Finish*.

   The Web Service Test utility opens, as shown in the following image.



SOAP 1.1

URL: http://localhost:8888/Hello

SOAP action: Hello

WSDL: http://localhost:8888/Hello?wsdl

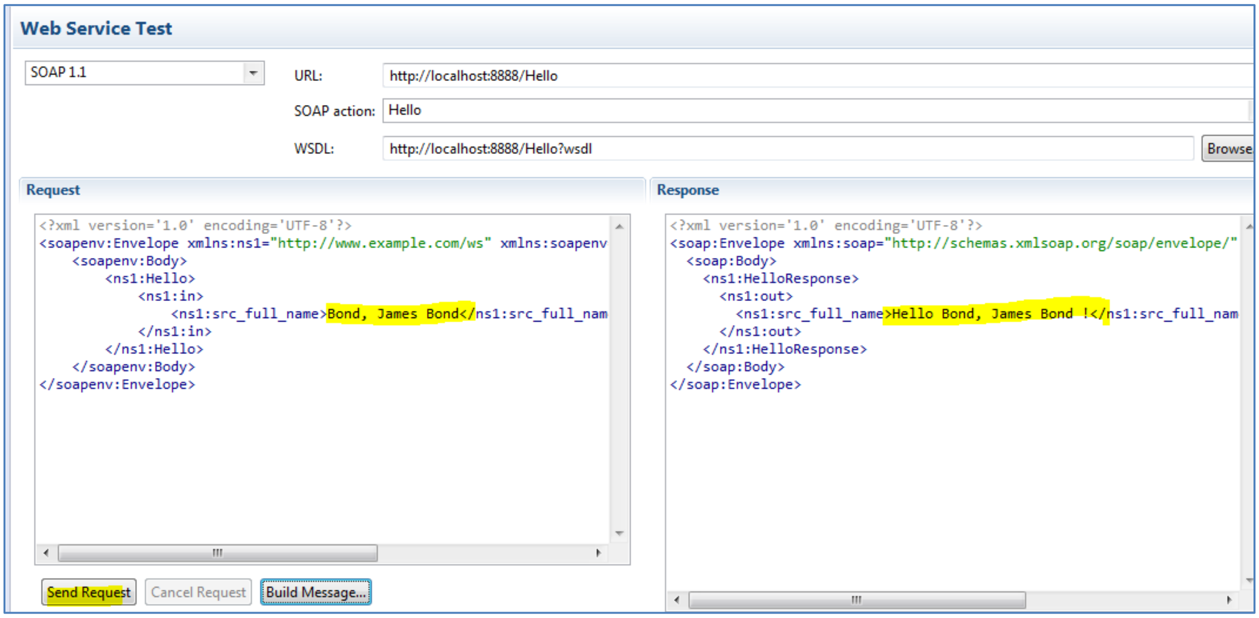
```
<?xml version='1.0' encoding='UTF-8'?>
<soapenv:Envelope xmlns:ns1="http://www.example.com/ws" xmlns:soapenv
    <soapenv:Body>
        <ns1:Hello>
            <ns1:in>
                <ns1:src_full_name>Bond, James Bond</ns1:src_full_nam
            </ns1:in>
        </ns1:Hello>
    </soapenv:Body>
</soapenv:Envelope>
```

```
<?xml version='1.0' encoding='UTF-8'?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  <soap:Body>
    <ns1:HelloResponse>
      <ns1:out>
        <ns1:src_full_name>Hello Bond, James Bond !</ns1:src_full_nam
      </ns1:out>
    </ns1:HelloResponse>
  </soap:Body>
</soap:Envelope>
```

A generic SOAP request is generated with sample values.

4. Enter a name in the SOAP request.

5. Click *Send Request* to test the web service, as shown in the following image.



SOAP 1.1

URL: http://localhost:8888/Hello

SOAP action: Hello

WSDL: http://localhost:8888/Hello?wsdl

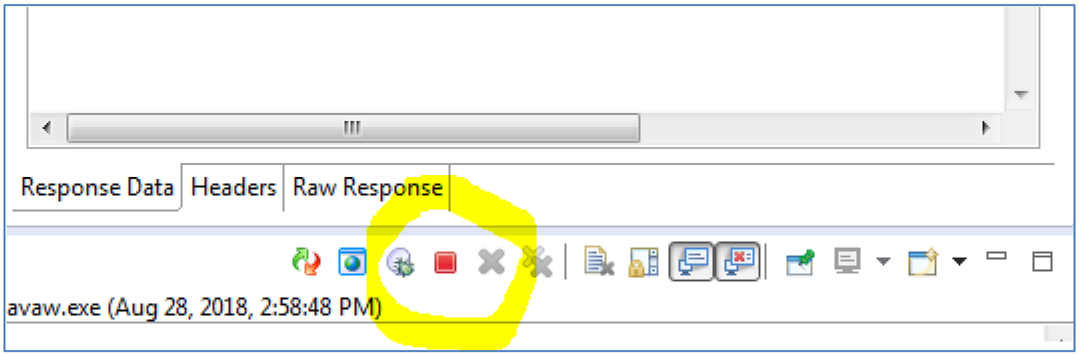
```
<?xml version='1.0' encoding='UTF-8'?>
<soapenv:Envelope xmlns:ns1="http://www.example.com/ws" xmlns:soapenv
    <soapenv:Body>
        <ns1:Hello>
            <ns1:in>
                <ns1:src_full_name>Bond, James Bond</ns1:src_full_nam
            </ns1:in>
        </ns1:Hello>
    </soapenv:Body>
</soapenv:Envelope>
```

```
<?xml version='1.0' encoding='UTF-8'?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  <soap:Body>
    <ns1:HelloResponse>
      <ns1:out>
        <ns1:src_full_name>Hello Bond, James Bond !</ns1:src_full_nam
      </ns1:out>
    </ns1:HelloResponse>
  </soap:Body>
</soap:Envelope>
```

6. View the Response to see the results of your DQ online service.

Stopping the Online Service

To stop the online service, click the red square icon, as shown in the following image.



Deploying the Online Service to the Runtime Engine

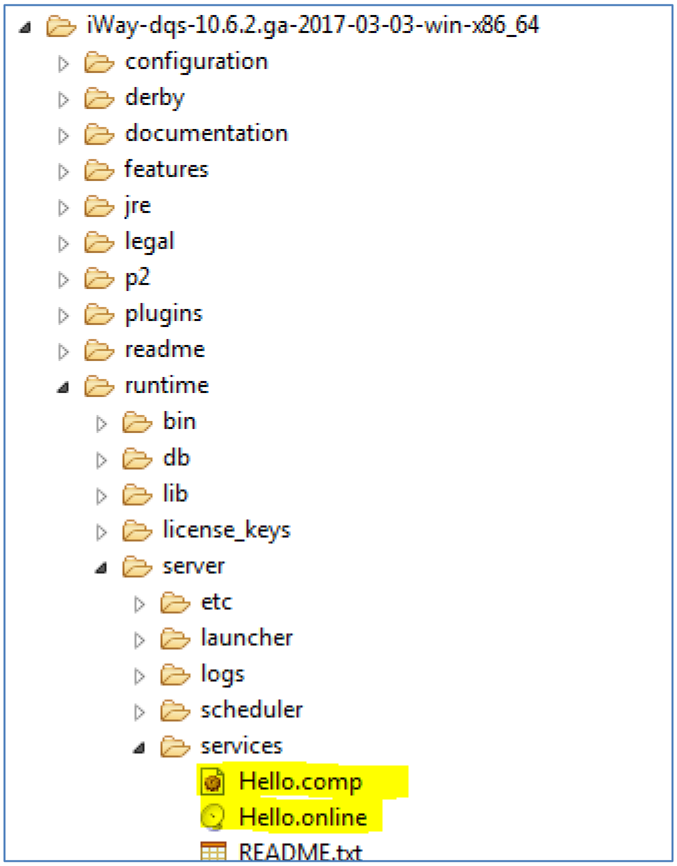
You can deploy your online service files to the runtime server, which could be a standalone DQ runtime environment or an iSM installation. In an iSM installation, the DQ runtime is stored in the following directory:

```
<iway_home>\etc\dqc
```

In a standalone DQ runtime environment, copy your .comp and .online files to the following directory:

```
<DQ_runtime>\server\services
```

For example:



```
iWay-dqs-10.6.2.ga-2017-03-03-win-x86_64
  configuration
  derby
  documentation
  features
  jre
  legal
  p2
  plugins
  readme
  runtime
    bin
    db
    lib
    license_keys
    server
      etc
      launcher
      logs
      scheduler
      services
        Hello.comp
        Hello.online
        README.txt
```

Starting the DQ Service

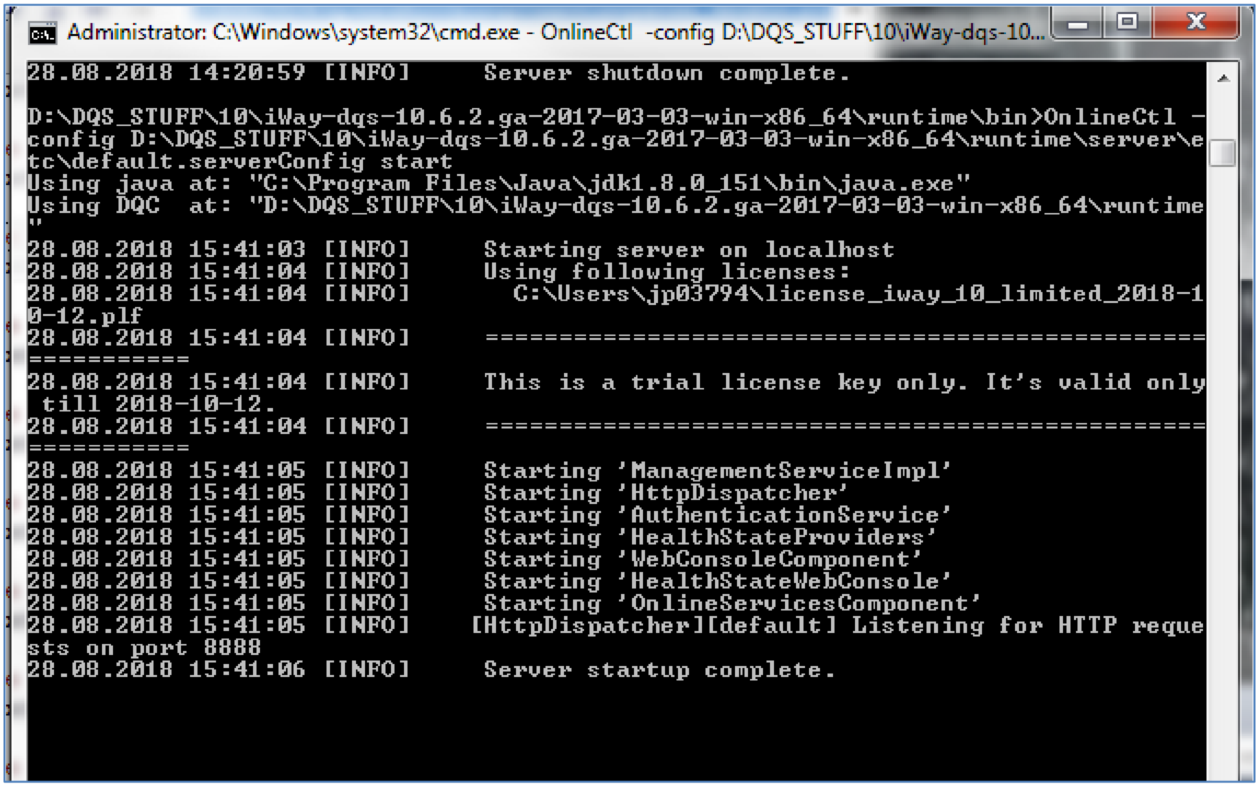
It is recommended to start the online service using the **OnlineCtl** command line tool, which provides parameters to start and stop the server.

1. Open a command prompt and enter the following commands:

   ```
   cd [DQ_HOME]\runtime\bin

   OnlineCtl -config [DQ_HOME]\runtime\server\etc\default.serverConfig start
   ```

   You can use the default configuration file (.serverConfig) if required. For more information on the .serverConfig file, see Appendix A, Understanding the .serverConfig Configuration File.



Administrator: C:\Windows\system32\cmd.exe - OnlineCtl -config D:\DQS_STUFF\10\iWay-dqs-10...

```
28.08.2018 14:20:59 [INFO]     Server shutdown complete.

D:\DQS_STUFF\10\iWay-dqs-10.6.2.ga-2017-03-03-win-x86_64\runtime\bin>OnlineCtl -
config D:\DQS_STUFF\10\iWay-dqs-10.6.2.ga-2017-03-03-win-x86_64\runtime\server\e
tc\default.serverConfig start
Using java at: "C:\Program Files\Java\jdk1.8.0_151\bin\java.exe"
Using DQC  at: "D:\DQS_STUFF\10\iWay-dqs-10.6.2.ga-2017-03-03-win-x86_64\runtime
"
28.08.2018 15:41:03 [INFO]     Starting server on localhost
28.08.2018 15:41:04 [INFO]     Using following licenses:
28.08.2018 15:41:04 [INFO]       C:\Users\jp03794\license_iway_10_limited_2018-1
0-12.plf
28.08.2018 15:41:04 [INFO]     =================================================
===========
28.08.2018 15:41:04 [INFO]     This is a trial license key only. It's valid only
 till 2018-10-12.
28.08.2018 15:41:04 [INFO]     =================================================
===========
28.08.2018 15:41:05 [INFO]     Starting 'ManagementServiceImpl'
28.08.2018 15:41:05 [INFO]     Starting 'HttpDispatcher'
28.08.2018 15:41:05 [INFO]     Starting 'AuthenticationService'
28.08.2018 15:41:05 [INFO]     Starting 'HealthStateProviders'
28.08.2018 15:41:05 [INFO]     Starting 'WebConsoleComponent'
28.08.2018 15:41:05 [INFO]     Starting 'HealthStateWebConsole'
28.08.2018 15:41:05 [INFO]     Starting 'OnlineServicesComponent'
28.08.2018 15:41:05 [INFO]    [HttpDispatcher][default] Listening for HTTP reque
sts on port 8888
28.08.2018 15:41:06 [INFO]     Server startup complete.
```

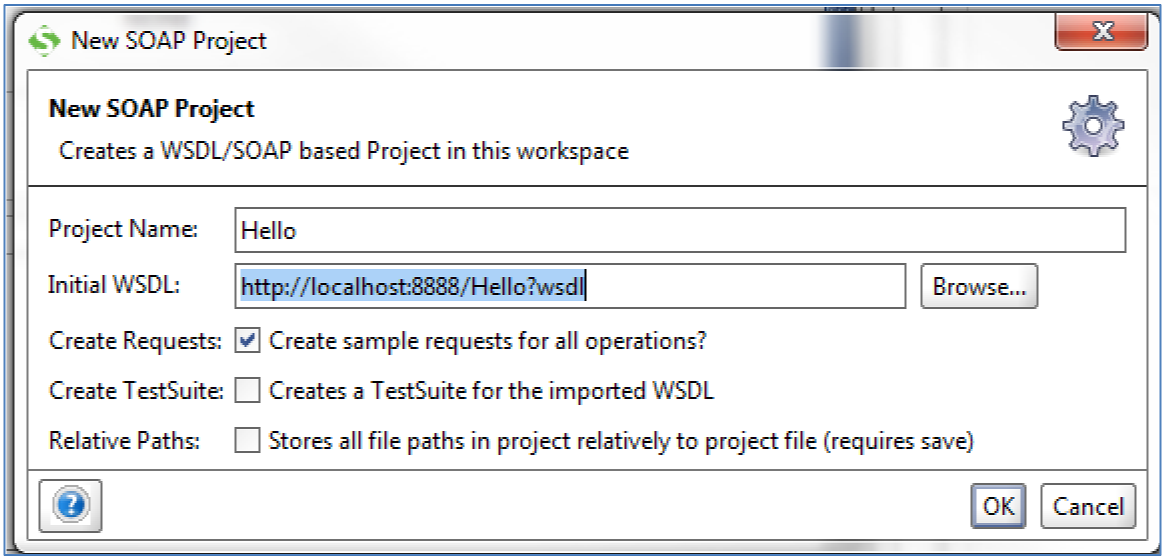
   Note: The default configuration file (.serverConfig) sets the service port to 8888. It also sets the DQ engine to use port 7777 for its own internal control.

Testing the Web Service From Any SOAP Client

This section describes how you can test your web service using the free SoapUI utility.

1. Create a new SOAP project using SoapUI.

2. Specify the WSDL URL in the Initial WSDL field, as shown in the following image.



New SOAP Project

Creates a WSDL/SOAP based Project in this workspace

Project Name: Hello

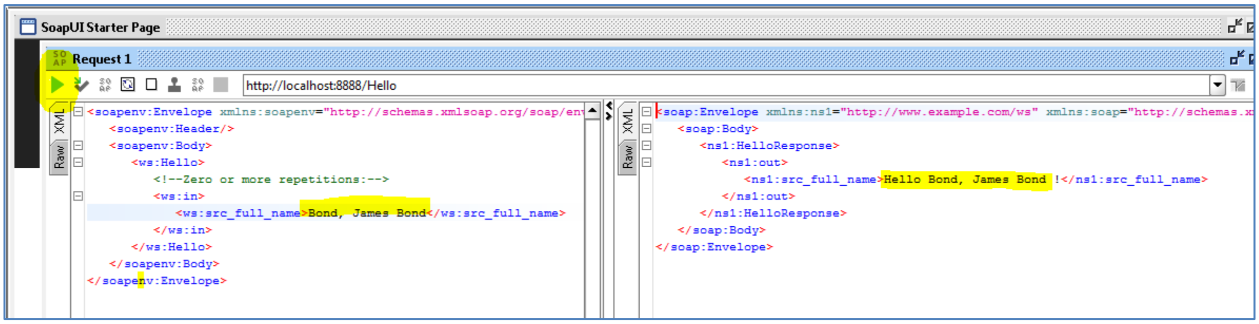
Initial WSDL: http://localhost:8888/Hello?wsdl

Create Requests: ☑ Create sample requests for all operations?

Create TestSuite: □ Creates a TestSuite for the imported WSDL

Relative Paths: □ Stores all file paths in project relatively to project file (requires save)

3. Enter a name in the SOAP request (*src_full_name* field) and execute the test, as shown in the following image.



http://localhost:8888/Hello

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/env
   <soapenv:Header/>
   <soapenv:Body>
      <ws:Hello>
         <!--Zero or more repetitions:-->
         <ws:in>
            <ws:src_full_name>Bond, James Bond</ws:src_full_name>
         </ws:in>
      </ws:Hello>
   </soapenv:Body>
</soapenv:Envelope>
```

```
<soap:Envelope xmlns:ns1="http://www.example.com/ws" xmlns:soap="http://schemas.x
   <soap:Body>
      <ns1:HelloResponse>
         <ns1:out>
            <ns1:src_full_name>Hello Bond, James Bond !</ns1:src_full_name>
         </ns1:out>
      </ns1:HelloResponse>
   </soap:Body>
</soap:Envelope>
```

Stopping and Restarting the DQ Service

During testing you may want to restart the DQ service often, whenever a change to the logic is made. It may be more convenient to do the following:

1. Leave the first OnlineCtl command session running, which started the DQ service.

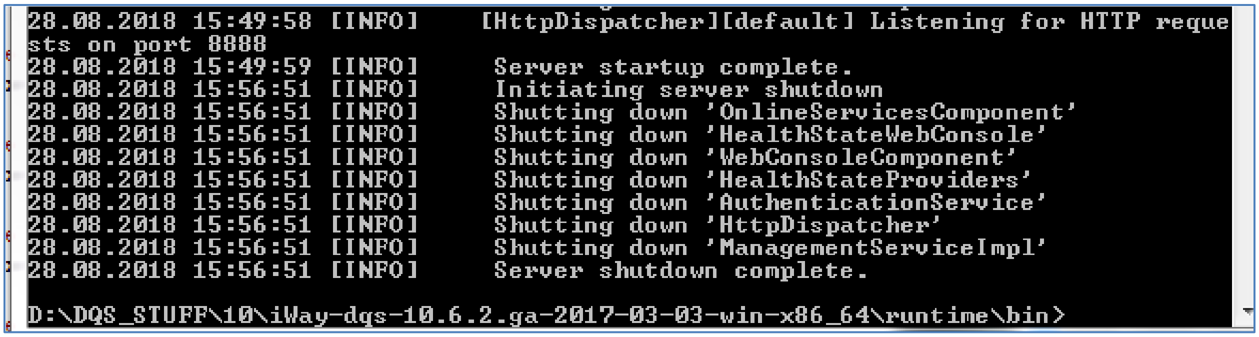
2. Start another OnlineCtl command session by opening a command prompt and entering the following:

   ```
   cd [DQ_HOME]\runtime\bin
   ```

3. Enter the following command to stop the DQ service:

   ```
   OnlineCtl -config [DQ_HOME]\runtime\server\etc\default.serverConfig stop
   ```

   Note: The configuration file (.serverConfig) is required to identify the correct port.



```
28.08.2018 15:49:58 [INFO]    [HttpDispatcher][default] Listening for HTTP reque
sts on port 8888
28.08.2018 15:49:59 [INFO]     Server startup complete.
28.08.2018 15:56:51 [INFO]     Initiating server shutdown
28.08.2018 15:56:51 [INFO]     Shutting down 'OnlineServicesComponent'
28.08.2018 15:56:51 [INFO]     Shutting down 'HealthStateWebConsole'
28.08.2018 15:56:51 [INFO]     Shutting down 'WebConsoleComponent'
28.08.2018 15:56:51 [INFO]     Shutting down 'HealthStateProviders'
28.08.2018 15:56:51 [INFO]     Shutting down 'AuthenticationService'
28.08.2018 15:56:51 [INFO]     Shutting down 'HttpDispatcher'
28.08.2018 15:56:51 [INFO]     Shutting down 'ManagementServiceImpl'
28.08.2018 15:56:51 [INFO]     Server shutdown complete.

D:\DQS_STUFF\10\iWay-dqs-10.6.2.ga-2017-03-03-win-x86_64\runtime\bin>
```

To restart the service, return to the first OnlineCtl command session, which should now indicate that the service has been stopped. Execute the OnlineCtl *Start* command.

Appendix A, Understanding the .serverConfig Configuration File

The server first reads the *.serverConfig* configuration file specified in the command line parameter. This file describes and starts the required server components. These components may use services from other components. Online services are provided by the *OnlineServicesComponent*:

```
<component class="com.ataccama.dqc.online.OnlineServicesComponent">
   <serviceLookupFolders>
             <configFolder>../services</configFolder>
  </serviceLookupFolders>
</component>
```

The *OnlineServicesComponent* contains the *configFolder* parameter, which points to the path on the file system containing the .online files. Each .online file defines a set of services (usually just one), which should be started.

The *OnlineServicesComponent* first copies all files (including the lookup files) from the *configFolder* to a private subfolder of the first temporary folder specified in the *.serverConfig* file. It then looks for all .online files and attempts to start all services.

If it is required to change a configuration or lookup files while the online service is running, it can list the folder that contains the configuration or lookup file in the *versionedFolders* section. Folders defined in the *versionedFolders* section of the *VersionedFileSystemComponent* will be treated similarly as the one defined by *configFolder,* except that they will not be scanned to find .online files. This means that these folders will be copied to a private subfolder of the temporary folder, and these files may be changed and refreshed.

If the process is successful, the service will run from the files copied in the temporary folder, so you can modify the configuration and/or change the lookup files. The services will respond without any interruption. When your configuration changes are ready, you should use the refresh command to update your services.

When the server is being stopped, it will first stop all of the services and then remove the temporary subfolder containing the previously copied configuration files. The original configuration remains untouched in the *configFolder* specified in the *.serverConfig* file.

The *OnlineServicesComponent* also requires the *HttpDispatcher* component to be started.