



# Denodo VQL Generation Stored Procedures - User Manual

Revision 20231027

## NOTE

This document is confidential and proprietary of **Denodo Technologies**. No part of this document may be reproduced in any form by any means without prior written authorization of **Denodo Technologies**.

Copyright © 2024  
Denodo Technologies Proprietary and Confidential

## CONTENTS

<b>1 OVERVIEW.....</b>	<b>9</b>
<b>2 DF.....</b>	<b>10</b>
<b>2.1 INSTALLATION.....</b>	<b>10</b>
<b>2.2 DESCRIPTION.....</b>	<b>11</b>
<b>2.3 SYNTAX.....</b>	<b>11</b>
<b>2.4 EXECUTION.....</b>	<b>13</b>
<b>2.4.1 EXECUTE AS A SELECT STATEMENT IN THE VQL SHELL WITHOUT TEMPLATE.....</b>	<b>13</b>
<b>2.4.1.1 SAMPLE AND RESULTS.....</b>	<b>13</b>
<b>2.4.2.1 SAMPLE AND RESULTS.....</b>	<b>15</b>
<b>2.5 .....</b>	<b>15</b>
<b>3 XML.....</b>	<b>16</b>
<b>3.1 INSTALLATION.....</b>	<b>16</b>
<b>3.2 GENERATE_VQL_TO_CREATE_XML_FULL_DATA_SOURCE_BASE_VIE W.....</b>	<b>17</b>
<b>3.2.1 DESCRIPTION.....</b>	<b>17</b>
<b>3.2.2 SYNTAX.....</b>	<b>17</b>
<b>3.3.1.1 EXECUTE AS A SELECT STATEMENT IN THE VQL SHELL WITHOUT TEMPLATE.....</b>	<b>19</b>
<b>3.3.1.1.1 SAMPLE AND RESULTS.....</b>	<b>19</b>
<b>3.3.1.2.1 SAMPLE AND RESULTS.....</b>	<b>21</b>

**3.5 GENERATE\_VQL\_TO\_CREATE\_XML\_DATA\_SOURCE\_BASE\_VIEW.....22**

**3.5.1 DESCRIPTION.....22**

**3.5.2 SYNTAX.....22**

**3.6.1.1.1 SAMPLE.....23**

**3.7 GENERATE\_VQL\_TO\_CREATE\_XML\_DATA\_SOURCE.....24**

**3.7.1 DESCRIPTION.....24**

**3.7.2 SYNTAX.....24**

**3.8.1.1.1 SAMPLE.....25**

**3.9 GENERATE\_VQL\_TO\_CREATE\_XML\_BASE\_VIEW.....26**

**3.9.1 DESCRIPTION.....26**

**3.9.2 SYNTAX.....26**

**3.10.1.1.1 SAMPLE.....27**

**4 JSON.....28**

**4.1 INSTALLATION.....28**

**4.2 GENERATE\_VQL\_TO\_CREATE\_JSON\_FULL\_DATA\_SOURCE\_BASE\_VIEW.....29**

**4.2.1 DESCRIPTION.....29**

**4.2.2 SYNTAX.....29**

**4.3.1.1 EXECUTE AS A SELECT STATEMENT IN THE VQL SHELL WITHOUT TEMPLATE.....31**

**4.3.1.1.1 SAMPLE AND RESULTS.....31**

- 4.3.1.2.1 SAMPLE AND RESULTS.....33**
  
- 4.5 GENERATE\_VQL\_TO\_CREATE\_JSON\_DATA\_SOURCE\_BASE\_VIEW  
.....34**

  - 4.5.1 DESCRIPTION.....34**
  - 4.5.2 SYNTAX.....34**

  
- 4.6.1.1 SAMPLE.....36**
  
- 4.7 GENERATE\_VQL\_TO\_CREATE\_JSON\_DATA\_SOURCE.....36**

  - 4.7.1 DESCRIPTION.....36**
  - 4.7.2 SYNTAX.....36**

  
- 4.8.1.1 SAMPLE.....38**
  
- 4.9 GENERATE\_VQL\_TO\_CREATE\_JSON\_BASE\_VIEW.....38**

  - 4.9.1 DESCRIPTION.....38**
  - 4.9.2 SYNTAX.....39**

  
- 4.10.1.1 SAMPLE.....39**
  
- 5 EXCEL.....41**
  - 5.1 INSTALLATION.....41**
  - 5.2 DESCRIPTION.....42**
  - 5.3 SYNTAX.....42**
    - 5.4 EXECUTION.....45**
  
- 5.4.1 EXECUTE AS A SELECT STATEMENT IN THE VQL SHELL  
WITHOUT TEMPLATE.....45**

  - 5.4.1.1 SAMPLE AND RESULTS.....45**

**5.4.2.1 SAMPLE AND RESULTS.....47**

**5.5 ..... 47**

**6 APPENDICES.....48**

**6.1.1 APPENDIX I. EXAMPLE OF S3 ROUTE EXECUTION.....48**

**6.1.3 RESULTS.....49**

## 1 OVERVIEW

---

This component contains three Stored Procedure that return the VQL statements required to create a Delimited File, XML, EXCEL or JSON Base View. These statements will include the creation of all components related with the base view such as data source, wrapper, complex types, tables and/or folder.

There are several procedures for each type of supported data source. It depends on the stored procedure type: you can create a data source, a base view, or both of them. You can also use an existing data source as a template from which all connection parameters will be copied.

*Note:* There is a similar stored procedure in VDP that returns the VQL statements required to create a JDBC base view. Follow this [link](#) for more information.

Also note that these procedures do not create the data sources or base views themselves, they just return the VQL statements to do so.

Denodo VQL Generation Stored Procedure version 20221006 or later requires VDP 8.0 update 20220815.  
If your VDP is earlier than 20220815 you can download the version 20220829 of Denodo VQL Generation Stored Procedure.

The only data types supported are LOCAL, HTTP Client and S3. This also applies if you use templates for the data routes.

## 2 DF

---

### 2.1 INSTALLATION

#### 2.1.1 Importing the Stored Procedure

In order to run the `GenerateVQLForDFFullDataSourceBaseViewStoredProcedure` stored procedure you have to load the `denodo-vqlgeneration.jar` file, using the `File > Jar Management` option of the VDP Administration Tool.

The class `GenerateVQLToCreateDFBaseViewStoredProcedure` is deprecated.

#### 2.1.2 Adding the Stored Procedure

##### 2.1.2.1 VQL Shell

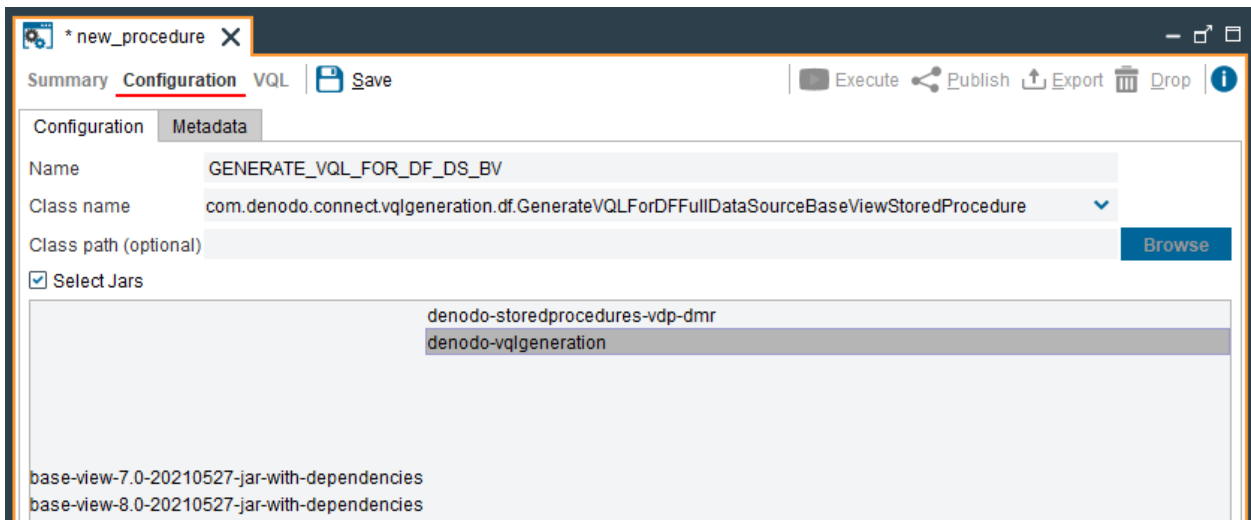
You can add the stored procedure programmatically with a `CREATE PROCEDURE` statement:

```
CREATE [OR REPLACE] PROCEDURE <name:identifier>  
CLASSNAME='com.denodo.connect.storedprocedure.df.GenerateVQLForDFFullDataSo  
urceBaseViewStoredProcedure'  
JARS 'denodo-vqlgeneration';  
  
[ FOLDER = <literal> ]  
[ DESCRIPTION = <literal> ]
```

##### 2.1.2.2 Virtual DataPort Administration Tool menu

You can also add the stored procedure by clicking on `Stored procedure` on the menu `File > New`:

You must input a name in the `Name` field and select the `Select Jars` checkbox in order to use the jar file, `denodo-vqlgeneration`, previously added to the Virtual DataPort Server.



## 2.2 DESCRIPTION

The GENERATE\_VQL\_TO\_CREATE\_DF\_DS\_BV stored procedure returns the VQL statements required to create a Delimited File base view from a delimited file. The stored procedure allows you to use an existing data source as a template to copy the configuration of the route, minus the field path or the url. Note that it does not actually create the view, it only returns the VQL statements to do so.

## 2.3 SYNTAX

```
GENERATE_VQL_TO_CREATE_DF_FULL_BASE_VIEW (
  data_source_name : text,
  route_type: text,
  file_path_url: text,
  authentication: text,
  username: text,
  password: text,
  encrypted_password: boolean,
  folder : text,
  database_name : text,
  i18n : text,
  base_view_name : text,
  base_view_folder: text,
  aws_access_key : text,
  aws_secret_key : text,
  aws_encrypted_secret_key : boolean,
  aws_iam_role_arn : text,
  use_aws_instance_credentials : boolean,
  aws_route_pattern : text,
  aws_custom_properties : text,
  template_datasource: text,
  template_database: text,
  column_delimiter:text,
  eol_delimiter:text,
  has_headers:text,
```



```
) ignore_matching_errors:text
)
```

- **data\_source\_name**: name of the data source. Mandatory.
- **route\_type**: specify the types of paths to access the file/s. Mandatory no template is specified.
  - LOCAL: the file or set of files to be used are located in the server machine.
  - HTTP: the data to be used has to be obtained by sending an HTTP request to a server.
  - S3: Use this type of route to obtain the data from a file or a set of files located in a S3 bucket.
- **file\_path\_url**: route to the delimited file or files. It can be a file path or a url or S3 path. Mandatory.
- **authentication**: authentication type (OFF|BASIC). Mandatory no template is specified.
- **username**: HTTP user.
- **password**: HTTP password.
- **encrypted\_password**: whether the password has been specified encrypted (as per Denodo VQL).
- **folder**: folder in which to place the data source, if the base view folder is not specified, the base view will be stored in the same folder. The result will include the VQL statements to create this folder(s). If null, the VQL will not specify a folder.
- **database\_name**: name of the database in which artifacts will be created. If null, the procedure uses the current database.
- **i18n**: i18n of the base view to be created. If null, the i18n of the current database will be assigned.
- **base\_view\_name**: name of the base view to be created. If null, the name will be auto-generated.
- **base\_view\_folder**: folder in which to place the base view. The result will include the VQL statements to create this folder(s). If null, the base view will be stored in the same folder as the datasource.
- **aws\_access\_key**: the AWS access key ID.
- **aws\_secret\_key**: the AWS secret key.
- **aws\_encrypted\_secret\_key**: whether the AWS secret key has been specified encrypted (as per Denodo VQL).
- **aws\_iam\_role\_arn**: Optionally, you can specify an AWS IAM role ARN to get the necessary privileges to connect to the database.
- **use\_aws\_instance\_credentials**: automatically obtain the credentials of the database from the AWS instance where this Virtual DataPort server is running.

- **aws\_route\_pattern**: Only for delimited-file data sources: if the path points to a directory and you enter a value in File name pattern, the data source will only process the files whose name matches the regular expression entered in this box. For example, if you only want to process the files with the extension log, enter `(.*)\.log`.
- **aws\_custom\_properties**: set the same properties that you would put in the Hadoop configuration files like `core-site.xml` to configure the S3A Hadoop connector.
- **template\_database**: name of the database where the template data source lives. If null and a template data source is specified, the current database will be assumed to contain it.
- **template\_datasource**: name of the data source from which the route configuration will be copied in the VQL. This parameter is mutually exclusive with `authentication`, `username`, `password` and `encrypted_password`. The route path will be modified by the field `file_path_url`.
- **ignore\_matching\_errors**: whether the wrapper should ignore lines of this data file(s) that do not have the expected structure (or else fail).
- **column\_delimiter**: character that separates the values of a row.
- **eol\_delimiter**: character string used as data tuple separator in the delimited file (`\n` line feeds will be used by default).
- **has\_headers**: If true, it is assumed that the first tuple extracted from the file data area contains the names of the fields. These names will be used to create the attributes of the base relation for Virtual DataPort.

## 2.4 EXECUTION

### 2.4.1 Execute as a SELECT statement in the VQL Shell without template

In order to obtain the VQL statements you have to specify the `datasource_name` and the `base_view_name`. And also choose the database where the base view would be stored, and a folder if it is the case.

All the parameters related with the datasource have to be introduced: `route_type`, `authentication` and `file_path_url`. Authentication parameters `username`, `password` and `encrypted_password` may be necessary also.

In addition, DF base views require the following parameters: `i18`, `column_delimiter`, `eol_delimite`, `has_headers` and `ignore_matching_errors`.

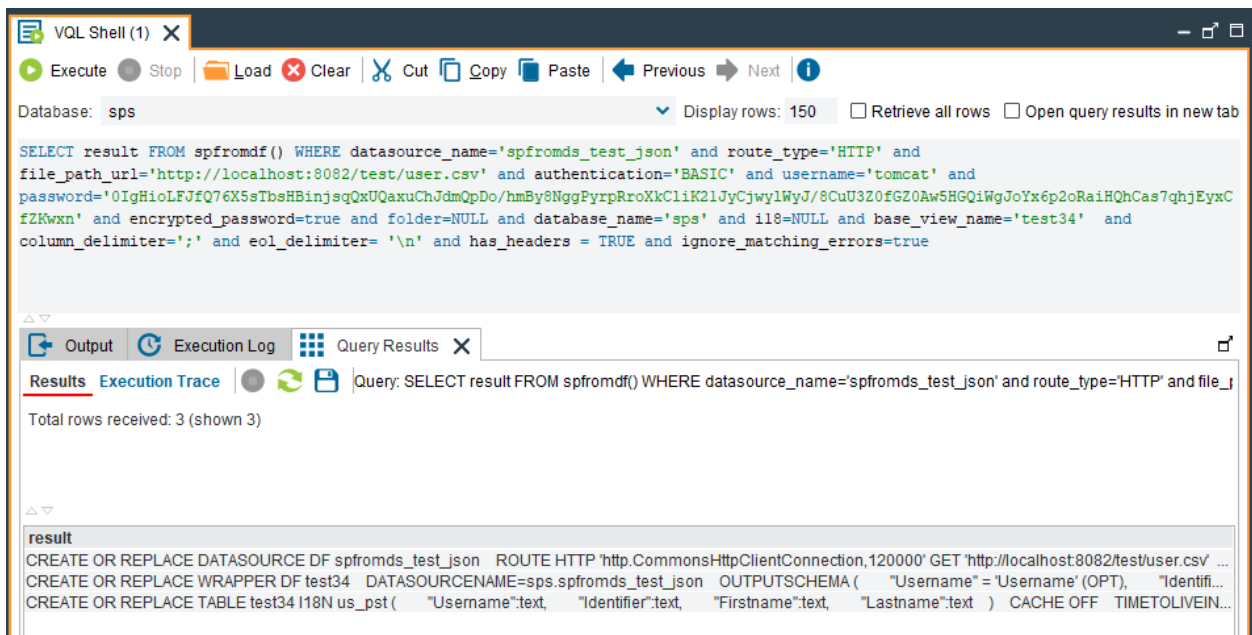
#### 2.4.1.1 Sample and results

```
SELECT result FROM GENERATE_VQL_TO_CREATE_DF_BASE_VIEW()  
WHERE datasource_name = 'gen_test_df'
```

```

and route_type = 'HTTP'
and file_path_url = 'http://localhost:8082/test/user.csv'
and authentication = 'BASIC'
and username = 'tomcat'
and password = '0IghioLFJfQ76X5sTbsHBinjsqQxUQas7qhjEyxCfZKwxn'
and encrypted_password = true
and folder = NULL
and database_name = 'sps'
and i18 = NULL
and base_view_name = 'test34'
and column_delimiter = ';'
and eol_delimiter= '\n'
and has_headers = TRUE
and ignore_matching_errors = true;

```



## 2.4.2 Execute as a SELECT statement in the VQL Shell using a template

In order to obtain the VQL statements you have to specify the `datasource_name` and the `base_view_name`. And also choose the database where the base view would be stored, and a folder if it is the case.

If you use another data source as a template in order to copy the route configuration you will need to introduce the name of the database (`template_database`) and the

name of the template data source (`template_datasource`). This way parameters `route_type`, `authentication`, `username`, `password` and `encrypted_password` will be imported from the template. But you have to introduce the `file_path_url` parameter, which is mandatory as the file path of the template data source will be overwritten. Note that despite using the template, parameters corresponding with the base view (and not the data source) are still mandatory: `i18`, `column_delimiter`, `eol_delimiter`, `has_headers` and `ignore_matching_errors`.

#### 2.4.2.1 Sample and results

```
SELECT result FROM GENERATE_VQL_TO_CREATE_DF_BASE_VIEW()
WHERE datasource_name = 'gen_test_df'
  and file_path_url = 'http://localhost:8082/test/user.csv'
  and folder = '/testds'
  and database_name = 'admin'
  and i18 = NULL
  and column_delimiter = ';'
  and eol_delimiter = '\n'
  and has_headers = TRUE
  and ignore_matching_errors = true
  and base_view_name = 'test49'
  and template_datasource = 'spfromds_test_df'
  and template_database = 'admin'
```

```
CREATE OR REPLACE DATASOURCE DF spfromds_test_df FOLDER = '/testds' ROUTE HTTP 'http.CommonsHttpClientConnection,120000' GET 'http://localhost:8...
CREATE OR REPLACE WRAPPER DF test49 FOLDER = '/testds' DATASOURCENAME=admin.spfromds_test_df OUTPUTSCHEMA ( "Username" = 'Usernam...
CREATE OR REPLACE TABLE test49 i18N us_pst ( "Username".text, "Identifier".text, "Firstname".text, "Lastname".text ) FOLDER = '/testds' CACHE ...
```

## 3 XML

---

There are 4 stored procedures to create XML data sources or XML base Views.

GenerateVQLForXMLFullDataSourceBaseViewStoredProcedure  
GenerateVQLForXMLDataSourceBaseViewStoredProcedure  
GenerateVQLForXMLDataSourceStoredProcedure  
GenerateVQLForXMLBaseViewStoredProcedure

The different stored procedures will be explained below.

### 3.1 **INSTALLATION**

#### 3.1.1 **Importing the Stored Procedure**

In order to run the GenerateVQLToCreateXMLBaseView stored procedure you have to load the denodo-vqlgeneration.jar file, using the File > Jar Management option of the VDP Administration Tool.

The class GenerateVQLToCreateXMLBaseViewStoredProcedure is deprecated.

#### 3.1.2 **Adding the Stored Procedure**

##### 3.1.2.1 VQL Shell

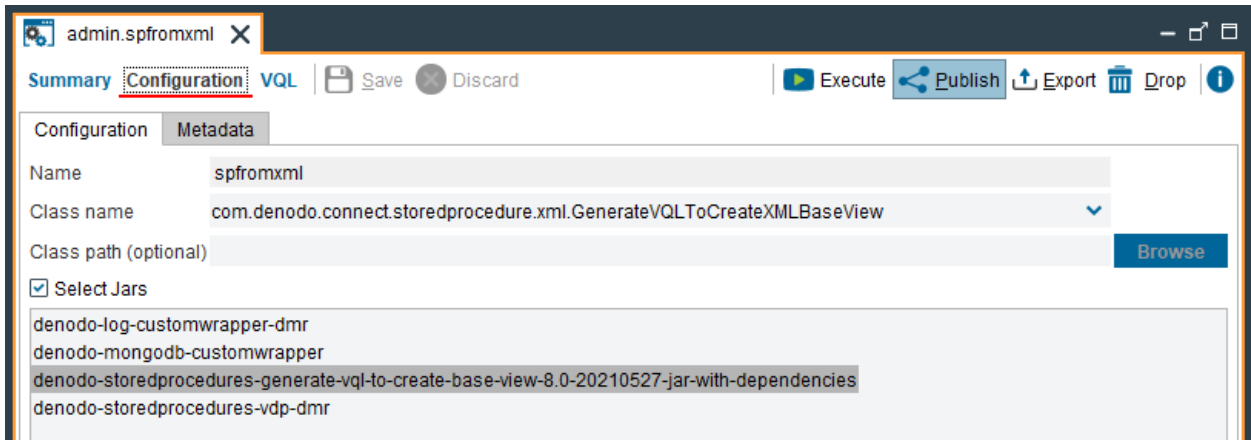
You can add the stored procedure programmatically with a CREATE PROCEDURE statement, you have to select the class, depending on the stored procedure you use.

```
CREATE [OR REPLACE] PROCEDURE <name:identifier>  
CLASSNAME='com.denodo.connect.storedprocedure.xml.GenerateVQLForXMLDataSour  
ceBaseViewStoredProcedure'  
JARS 'denodo-vqlgeneration';  
  
[ FOLDER = <literal> ]  
[ DESCRIPTION = <literal> ]
```

##### 3.1.2.2 Virtual DataPort Administration Tool menu

You can also add the stored procedure by clicking on Stored procedure on the menu File > New:

You must input a name in the Name field and select the Select Jars checkbox in order to use the Jar file, denodo-vqlgeneration, previously added to the Virtual DataPort Server. And select the Class name in the drop-down.



## 3.2 GENERATE\_VQL\_TO\_CREATE\_XML\_FULL\_DATA\_SOURCE\_BASE\_VIEW

### 3.2.1 Description

The `GENERATE_VQL_TO_CREATE_XML_FULL_DATA_SOURCE_BASE_VIEW` stored procedure returns the VQL statements required to create a XML data source and a XML base view from an XML file. The stored procedure allows you to use an existing data source as a template to copy the configuration of the route, minus the field path or the url. Note that it does not actually create the view, it only returns the VQL statements to do so.

### 3.2.2 Syntax

## 3.3

```
GENERATE_VQL_TO_CREATE_XML_FULL_DATA_SOURCE_BASE_VIEW (
  data_source_name : text,
  route_type: text,
  file_path_url: text,
  authentication: text,
  username: text,
  password: text,
  encrypted_password: text,
  folder : text,
  database_name : text,
  i18n : text,
  base_view_name : text,
  base_view_folder : text,
  aws_access_key : text,
  aws_secret_key : text,
  aws_encrypted_secret_key : boolean,
  aws_iam_role_arn : text,
  use_aws_instance_credentials : boolean,
  aws_route_pattern : text,
  aws_custom_properties : text,
  template_datasource: text,
  template_database: text
```

)

- **data\_source\_name**: name of the data source. Mandatory.
- **route\_type**: specify the types of paths to access the file/s. Mandatory no template is specified.
  - LOCAL: the file or set of files to be used are located in the server machine.
  - HTTP: the data to be used has to be obtained by sending an HTTP request to a server.
  - S3: Use this type of route to obtain the data from a file or a set of files located in a S3 bucket.
- **file\_path\_url**: route to the delimited file or files. It can be a file path or a url. If the route\_type is http ,it is data source's URL, it can be base url or absolute url including the filename. Mandatory.
- **authentication**: authentication type (OFF|BASIC). Mandatory no template is specified.
- **username**: HTTP user.
- **password**: HTTP password.
- **encrypted\_password**: whether the password has been specified encrypted (as per Denodo VQL).
- **folder**: folder in which to place the data source, if the base view folder is not specified, the base view will be stored in the same folder. The result will include the VQL statements to create this folder(s). If null, the VQL will not specify a folder.
- **database\_name**: name of the database in which artifacts will be created. If null, the procedure uses the current database.
- **i18n**: i18n of the base view to be created. If null, the i18n of the current database will be assigned.
- **base\_view\_name**: name of the base view to be created. If null, the name will be auto-generated.
- **base\_view\_folder**: folder in which to place the base view. The result will include the VQL statements to create this folder(s). If null, the base view will be stored in the same folder as the datasource.
- **aws\_access\_key**: the AWS access key ID.
- **aws\_secret\_key**: the AWS secret key.
- **aws\_encrypted\_secret\_key**: whether the AWS secret key has been specified encrypted (as per Denodo VQL).

- **aws\_iam\_role\_arn**: Optionally, you can specify an AWS IAM role ARN to get the necessary privileges to connect to the database.
- **use\_aws\_instance\_credentials**: automatically obtain the credentials of the database from the AWS instance where this Virtual DataPort server is running.
- **aws\_route\_pattern**: Only for delimited-file data sources: if the path points to a directory and you enter a value in File name pattern, the data source will only process the files whose name matches the regular expression entered in this box. For example, if you only want to process the files with the extension log, enter `(.*)\.log`.
- **aws\_custom\_properties**: set the same properties that you would put in the Hadoop configuration files like `core-site.xml` to configure the S3A Hadoop connector.
- 
- **template\_database**: name of the database where the template data source lives. If null and a template data source is specified, the current database will be assumed to contain it.
- **template\_datasource**: name of the data source from which the route configuration will be copied in the VQL. This parameter is mutually exclusive with `authentication`, `username`, `password` and `encrypted_password`. The route path will be modified by the field `file_path_url`.

### 3.3.1 Execution

#### 3.3.1.1 Execute as a SELECT statement in the VQL Shell without template

In order to obtain the VQL statements you have to specify the `datasource_name` and the `base_view_name`. And also choose the database where the base view would be stored, and a folder if it is the case.

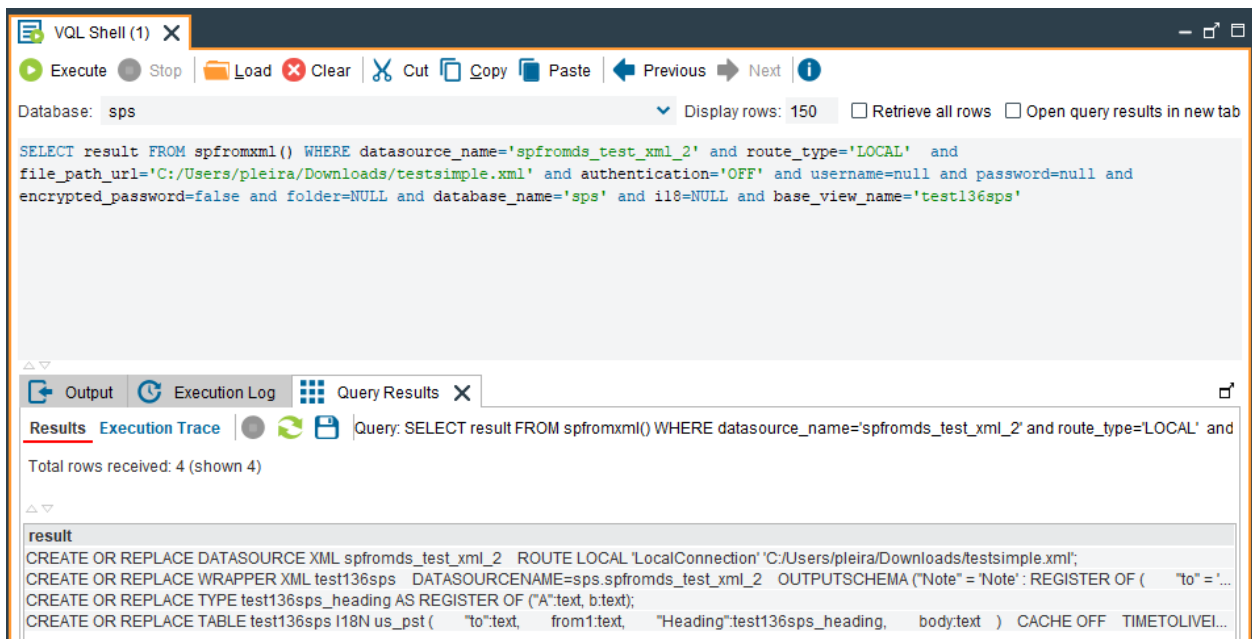
All the parameters related with the `datasource` have to be introduced: `route_type`, `authentication` and `file_path_url`. Authentication parameters `username`, `password` and `encrypted_password` may be necessary also.

##### 3.3.1.1.1 Sample and results

```
SELECT result FROM GENERATE_VQL_TO_CREATE_XML_FULL_DATA_SOURCE_BASE_VIEW()  
WHERE datasource_name = 'gen_test_xml'  
and route_type = 'LOCAL'  
and file_path_url = 'C:/Users/denodo/Downloads/testsimple.xml'
```



```
and authentication = 'OFF'
and username = null
and password = null
and encrypted_password = false
and folder = NULL
and database_name = 'sps'
and i18 = NULL
and base_view_name = 'test136sps'
```



### 3.3.1.2 Execute as a SELECT statement in the VQL Shell using a template

In order to obtain the VQL statements you have to specify the `datasource_name` and the `base_view_name`. And also choose the database where the base view would be stored, and a folder if it is the case.

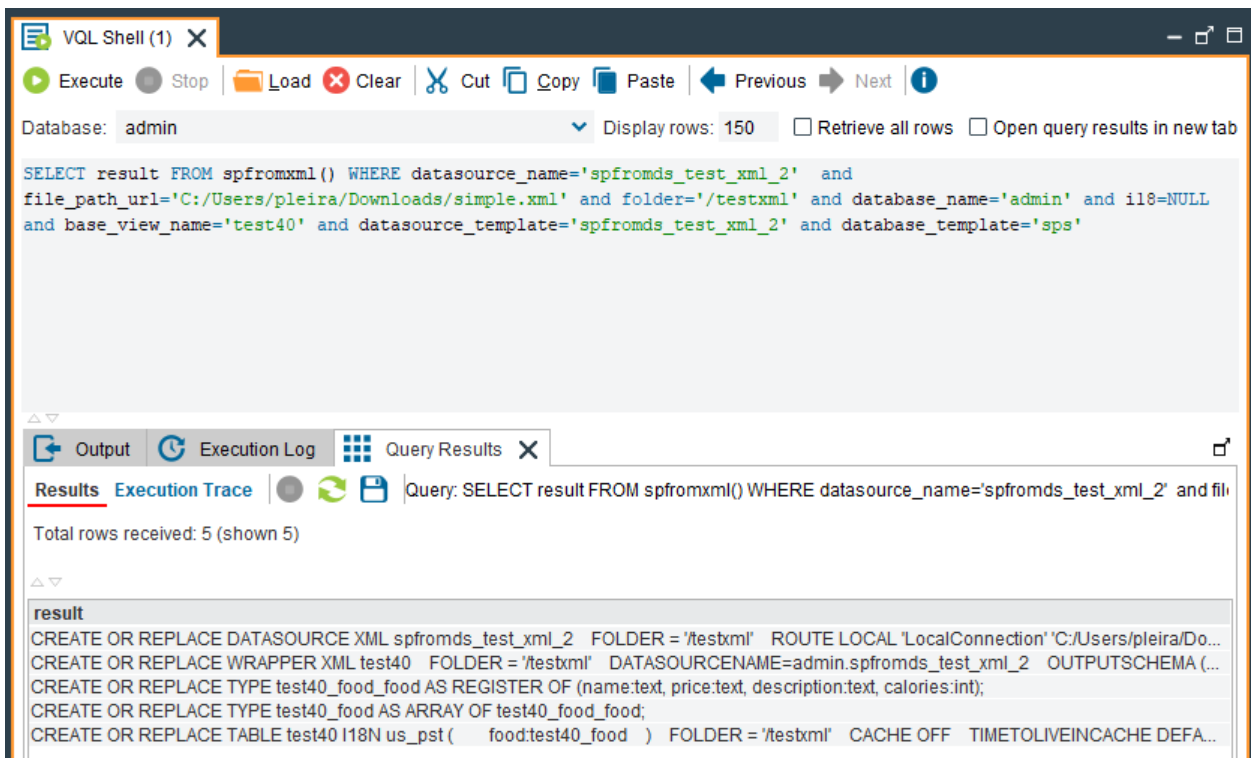
If you use another data source as a template in order to copy the route configuration you will need to introduce the name of the database (`template_database`) and the name of the template data source (`template_datasource`). This way parameters `route_type`, `authentication`, `username`, `password` and `encrypted_password` will be imported from the template. But you have to introduce the `file_path_url` parameter, which is mandatory as the file path of the template data source will be overwritten.

The `i18` parameter is also mandatory.

### 3.3.1.2.1 Sample and results

```
SELECT result FROM GENERATE_VQL_TO_CREATE_XML_FULL_DATA_SOURCE_BASE_VIEW()
WHERE datasource_name = 'gen_test_xml1'
and file_path_url = 'C:/Users/denodo/Downloads/simple.xml1'
and folder = '/testxml1'
and database_name = 'admin'
and i18 = NULL
and base_view_name = 'test40'
and template_datasource = 'spfromds_test_xml_2'
and template_database = 'sps'
```

## 3.4



The screenshot shows the VQL Shell interface. The top toolbar includes buttons for Execute, Stop, Load, Clear, Cut, Copy, Paste, Previous, Next, and an information icon. The Database is set to 'admin'. The query editor contains the following SQL query:

```
SELECT result FROM spfromxml() WHERE datasource_name='spfromds_test_xml_2' and
file_path_url='C:/Users/pleira/Downloads/simple.xml1' and folder='/testxml1' and database_name='admin' and i18=NULL
and base_view_name='test40' and datasource_template='spfromds_test_xml_2' and database_template='sps'
```

The Results tab is active, showing the execution output. The query executed is: `SELECT result FROM spfromxml() WHERE datasource_name='spfromds_test_xml_2' and file...`. The output shows 5 rows received. The first row is:

```
CREATE OR REPLACE DATASOURCE XML spfromds_test_xml_2 FOLDER = '/testxml1' ROUTE LOCAL 'LocalConnection' 'C:/Users/pleira/Do...
```

Other rows in the result set include:

```
CREATE OR REPLACE WRAPPER XML test40 FOLDER = '/testxml1' DATASOURCENAME=admin.spfromds_test_xml_2 OUTPUTSCHEMA (...
CREATE OR REPLACE TYPE test40_food_food AS REGISTER OF (name:text, price:text, description:text, calories:int);
CREATE OR REPLACE TYPE test40_food AS ARRAY OF test40_food_food;
CREATE OR REPLACE TABLE test40 I18N us_pst ( food:test40_food ) FOLDER = '/testxml1' CACHE OFF TIMETOLIVEINCACHE DEFA...
```

## 3.5 GENERATE\_VQL\_TO\_CREATE\_XML\_DATA\_SOURCE\_BASE\_VIEW

### 3.5.1 Description

The `GENERATE_VQL_TO_CREATE_XML_DATA_SOURCE_BASE_VIEW` stored procedure returns the VQL statements required to create a XML data source and a XML base view from an XML file. With this stored procedure if the `route_type` is HTTP, using `relative_url` parameter, we can differentiate the data source URL and base view URL, the base URL(data source) will be concatenated to base view's relative URL. Note that it does not actually create the view, it only returns the VQL statements to do so. This stored procedure requires VDP 8.0 update 20220815 or later.

### 3.5.2 Syntax

## 3.6

```
GENERATE_VQL_TO_CREATE_XML_DTA_SOURCE_BASE_VIEW (  
  data_source_name : text,  
  route_type: text,  
  file_path_url: text,  
  authentication: text,  
  username: text,  
  password: text,  
  encrypted_password: text,  
  folder : text,  
  database_name : text,  
  i18n : text,  
  base_view_name : text,  
  base_view_folder : text  
)
```

- ***data\_source\_name***: name of the data source. Mandatory.
- ***database\_name***: name of the database in which artifacts will be created. If null, the procedure uses the current database.
- ***base\_view\_name***: name of the base view to be created. If null, the name will be auto-generated.
- ***route\_type***: specify the types of paths to access the file/s. Mandatory no template is specified.
  - LOCAL: the file or set of files to be used are located in the server machine.
  - HTTP: the data to be used has to be obtained by sending an HTTP request to a server.
- ***file\_path\_url***: route to the delimited file or files. It can be a file path or a url. If the `route_type` is http ,it is data source's URL, it can be base url or absolute url including the filename. Mandatory.

- **relative\_url**: Only it is used when the route\_type is HTTP, and it is optional. Base view's relative URL (for example: "/books.xml"), this URL will be concatenated to the Base URL when executing the base view (the resulting URL will be http://www.test.com/books.xml).
- **authentication**: authentication type (OFF|BASIC). Mandatory no template is specified.
- **username**: HTTP user.
- **password**: HTTP password.
- **encrypted\_password**: whether the password has been specified encrypted (as per Denodo VQL).
- **folder**: folder in which to place the data source, if the base view folder is not specified, the base view will be stored in the same folder. The result will include the VQL statements to create this folder(s). If null, the VQL will not specify a folder.
- **i18n**: i18n of the base view to be created. If null, the i18n of the current database will be assigned.
- **base\_view\_folder**: folder in which to place the base view. The result will include the VQL statements to create this folder(s). If null, the base view will be stored in the same folder as the datasource.

### 3.6.1 Execution

In order to obtain the VQL statements you have to specify the datasource\_name and the base\_view\_name. And also choose the database where the base view would be stored, and a folder if it is the case.

All the parameters related with the datasource have to be introduced: route\_type, authentication and file\_path\_url. Authentication parameters username, password and encrypted\_password may be necessary also.

Belonging to the base view there is the optional i18n parameter. And if the data source has a base URL, relative\_url parameter allows having a relative URL in the base view.

#### 3.6.1.1.1 Sample

```
SELECT result FROM generate_vql_to_create_xml_data_source_base_view()
```

```
WHERE datasource_name='test_xml_datasource'  
and route_type='HTTP'  
and file_path_url='https://www.w3schools.com/xml'  
and authentication='OFF'  
and folder='/testsd'  
and database_name='admin'  
and base_view_name='test_xml_datasource_all_base_view'  
and relative_url='cd_catalog.xml'
```

## 3.7 GENERATE\_VQL\_TO\_CREATE\_XML\_DATA\_SOURCE

### 3.7.1 Description

The `GENERATE_VQL_TO_CREATE_XML_DATA_SOURCE` stored procedure returns the VQL statements required to create a XML data source from an XML file. Note that it does not actually create the view, it only returns the VQL statements to do so. This stored procedure requires VDP 8.0 update 20220815 or later.

### 3.7.2 Syntax

## 3.8

```
GENERATE_VQL_TO_CREATE_XML_DATA_SOURCE (  
  data_source_name : text,  
  database_name : text,  
  route_type: text,  
  file_path_url: text,  
  authentication: text,  
  username: text,  
  password: text,  
  encrypted_password: text,  
  base_view_name : text,  
  base_view_folder : text,  
  aws_access_key : text,  
  aws_secret_key : text,  
  aws_encrypted_secret_key : boolean,  
  aws_iam_role_arn : text,  
  use_aws_instance_credentials : boolean,  
  aws_route_pattern : text,  
  folder : text  
)
```

- ***data\_source\_name***: name of the data source. Mandatory.
- ***database\_name***: name of the database in which artifacts will be created. If null, the procedure uses the current database.

- **route\_type**: specify the types of paths to access the file/s. Mandatory no template is specified.
  - LOCAL: the file or set of files to be used are located in the server machine.
  - HTTP: the data to be used has to be obtained by sending an HTTP request to a server.
  - S3: Use this type of route to obtain the data from a file or a set of files located in a S3 bucket.
- **file\_path\_url**: route to the delimited file or files. It can be a file path or a url. If the route\_type is http ,it is data source's URL, it can be base url or absolute url including the filename. Mandatory.
- **authentication**: authentication type (OFF|BASIC). Mandatory no template is specified.
- **username**: HTTP user.
- **password**: HTTP password.
- **encrypted\_password**: whether the password has been specified encrypted (as per Denodo VQL).
- **folder**: folder in which to place the created structures. The result will include the VQL statements to create this folder(s). If null, the VQL will not specify a folder.

### 3.8.1 Execution

In order to obtain the VQL statements you have to specify the `datasource_name`. And also choose the database where the base view would be stored, and a folder if it is the case.

All the parameters related with the datasource have to be introduced: `route_type`, `authentication` and `file_path_url`. Authentication parameters `username`, `password` and `encrypted_password` may be necessary also.

#### 3.8.1.1.1 Sample

```
SELECT result FROM generate_vql_to_create_xml_data_source()  
WHERE datasource_name='test_xml_datasource'  
and route_type='HTTP'  
and file_path_url='https://www.w3schools.com/xml'  
and authentication='OFF'  
and folder='/testsd'  
and database_name='admin'
```

## **3.9 GENERATE\_VQL\_TO\_CREATE\_XML\_BASE\_VIEW**

### **3.9.1 Description**

The `GENERATE_VQL_TO_CREATE_XML_BASE_VIEW` stored procedure returns the VQL statements required to create an XML base view from an XML file, over a existing data source. If the route type of the data source is HTTP, and the path is not a complete URL, the `relative_url` would be added to the base URL. Note that it does not actually create the view, it only returns the VQL statements to do so. This stored procedure requires VDP 8.0 update 20220815 or later.

### **3.9.2 Syntax**

### **3.10**

```
GENERATE_VQL_TO_CREATE_XML_BASE_VIEW (  
  data_source_name : text,  
  i18n : text,  
  database_name : text,  
  base_view_name : text,  
  base_view_folder : text,  
  relative_url: text,  
  
)
```

- ***data\_source\_name***: name of the data source. Mandatory.
- ***database\_name***: name of the database in which artifacts will be created. If null, the procedure uses the current database.
- ***base\_view\_name***: name of the base view to be created. If null, the name will be auto-generated.
- ***i18n***: i18n of the base view to be created. If null, the i18n of the current database will be assigned.
- ***relative\_url***: Only it is used when the `route_type` is HTTP, and it is optional. Base view's relative URL (for example: `"/books.json"`), this URL will be concatenated to the Data Source URL when executing the base view.
- ***base\_view\_folder***: folder in which to place the base view. The result will include the VQL statements to create this folder(s). If null, the base view will be stored in the same folder as the datasource.

### 3.10.1 Execution

#### 3.10.1.1

In order to obtain the VQL statements you have to specify the `datasource_name` and the `base_view_name`. And also choose the database where the data source is stored. In addition there is the optional `i18n` parameter.

##### 3.10.1.1.1 Sample

```
SELECT result FROM GENERATE_VQL_TO_CREATE_XML_BASE_VIEW()  
WHERE datasource_name='test_xml_datasource'  
and database_name='admin'  
and base_view_name='test_xml_datasource_bv'  
and relative_url='cd_catalog.xml'
```



## 4 JSON

---

### 4.1 INSTALLATION

#### 4.1.1 Importing the Stored Procedure

In order to run the `GenerateVQLToCreateJSONBaseView` stored procedure you have to load the `denodo-vqlgeneration.jar` file, using the `File > Jar Management` option of the `VDP Administration Tool`. The class `GenerateVQLToCreateJSONBaseViewStoredProcedure` is deprecated.

#### 4.1.2 Adding the Stored Procedure

##### 4.1.2.1 VQL Shell

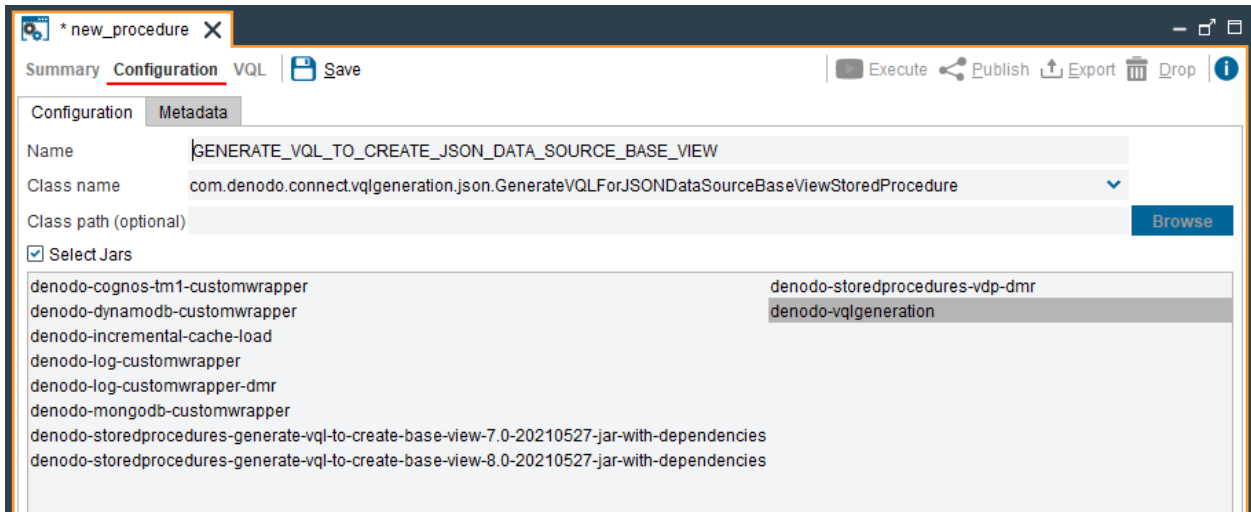
You can add the stored procedure programmatically with a `CREATE PROCEDURE` statement, you have to select the class, depending on the stored procedure you use.

```
CREATE [OR REPLACE] PROCEDURE <name:identifier>  
CLASSNAME='com.denodo.connect.storedprocedure.json.GenerateVQLForJSONDataSo  
urceBaseViewStoredProcedure'  
JARS 'denodo-vqlgeneration';  
  
[ FOLDER = <literal> ]  
[ DESCRIPTION = <literal> ]
```

##### 4.1.2.2 Virtual DataPort Administration Tool menu

You can also add the stored procedure by clicking on `Stored procedure` on the menu `File > New`:

You must input a name in the `Name` field and select the `Select Jars` checkbox in order to use the jar file, `denodo-vqlgeneration`, previously added to the `Virtual DataPort Server`. And select the `Class` name that you need, in the drop-down.



## **4.2 GENERATE\_VQL\_TO\_CREATE\_JSON\_FULL\_DATA\_SOURCE\_BASE\_VIEW**

### **4.2.1 Description**

The `GENERATE_VQL_TO_CREATE_JSON_FULL_DATA_SOURCE_BASE_VIEW` stored procedure returns the VQL statements required to create a JSON data source and a JSON base view from a JSON file. The stored procedure allows you to use an existing data source as a template to copy the configuration of the route, minus the field path or the url. Note that it does not actually create the view, it only returns the VQL statements to do so.

### **4.2.2 Syntax**

### **4.3**

```
GENERATE_VQL_TO_CREATE_JSON_FULL_DATA_SOURCE_BASE_VIEW (
  data_source_name : text,
  route_type: text,
  file_path_url: text,
  authentication: text,
  username: text,
  password: text,
  encrypted_password: text,
  folder : text,
  database_name : text,
  i18n : text,
  base_view_name : text,
  base_view_folder _ text,
  aws_access_key : text,
  aws_secret_key : text,
  aws_encrypted_secret_key : boolean,
  aws_iam_role_arn : text,
```

```

use_aws_instance_credentials : boolean,
aws_route_pattern : text,
aws_custom_properties : text,
template_datasource: text,
template_database: text
)

```

- **data\_source\_name**: name of the data source. Mandatory.
- **database\_name**: name of the database in which artifacts will be created. If null, the procedure uses the current database.
- **base\_view\_name**: name of the base view to be created. If null, the name will be auto-generated.
- **route\_type**: specify the types of paths to access the file/s. Mandatory no template is specified.
  - LOCAL: the file or set of files to be used are located in the server machine.
  - HTTP: the data to be used has to be obtained by sending an HTTP request to a server.
  - S3: Use this type of route to obtain the data from a file or a set of files located in a S3 bucket.
- **file\_path\_url**: route to the delimited file or files. It can be a file path or a url. If the route\_type is http ,it is data source's URL, it can be base url or absolute url including the filename. Mandatory.
- **authentication**: authentication type (OFF|BASIC). Mandatory no template is specified.
- **username**: HTTP user.
- **password**: HTTP password.
- **encrypted\_password**: whether the password has been specified encrypted (as per Denodo VQL).
- **folder**: folder in which to place the data source, if the base view folder is not specified, the base view will be stored in the same folder. The result will include the VQL statements to create this folder(s). If null, the VQL will not specify a folder.
- **i18n**: i18n of the base view to be created. If null, the i18n of the current database will be assigned.
- **base\_view\_folder**: folder in which to place the base view. The result will include the VQL statements to create this folder(s). If null, the base view will be stored in the same folder as the datasource.
- **aws\_access\_key**: the AWS access key ID.
- **aws\_secret\_key**: the AWS secret key.
- **aws\_encrypted\_secret\_key**: whether the AWS secret key has been specified encrypted (as per Denodo VQL).
- **aws\_iam\_role\_arn**: Optionally, you can specify an AWS IAM role ARN to get the necessary privileges to connect to the database.

- ***use\_aws\_instance\_credentials***: automatically obtain the credentials of the database from the AWS instance where this Virtual DataPort server is running.
- ***aws\_route\_pattern***: Only for delimited-file data sources: if the path points to a directory and you enter a value in File name pattern, the data source will only process the files whose name matches the regular expression entered in this box. For example, if you only want to process the files with the extension log, enter `(.*)\.log`.
- ***aws\_custom\_properties***: set the same properties that you would put in the Hadoop configuration files like `core-site.xml` to configure the S3A Hadoop connector.
- 
- ***template\_database***: name of the database where the template data source lives. If null and a template data source is specified, the current database will be assumed to contain it.
- ***template\_datasource***: name of the data source from which the route configuration will be copied in the VQL. This parameter is mutually exclusive with authentication, username, password and `encrypted_password`. The route path will be modified by the field `file_path_url`.

### 4.3.1 Execution

#### 4.3.1.1 Execute as a SELECT statement in the VQL Shell without template

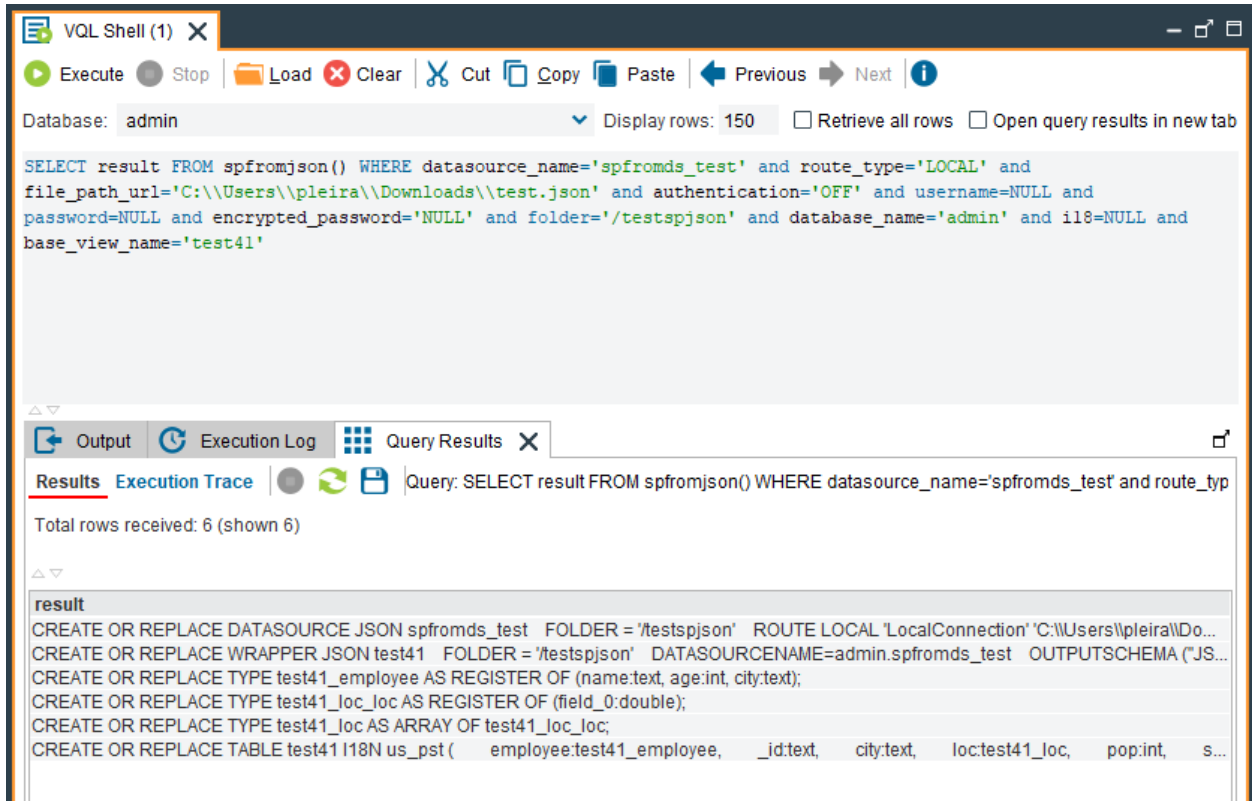
In order to obtain the VQL statements you have to specify the `datasource_name` and the `base_view_name`. And also choose the database where the base view would be stored, and a folder if it is the case.

All the parameters related with the datasource have to be introduced: `route_type`, authentication and `file_path_url`. Authentication parameters `username`, `password` and `encrypted_password` may be necessary also.

##### 4.3.1.1.1 Sample and results

```
SELECT result FROM GENERATE_VQL_TO_CREATE_JSON_DATA_SOURCE_BASE_VIEW()  
WHERE datasource_name='gen_test_json'  
and route_type='LOCAL'  
and file_path_url='C:/Users/denodo/Downloads/test.json'  
and authentication='OFF'  
and username=NULL  
and password=NULL
```

```
and encrypted_password=false
and folder='/testspjson'
and database_name='admin'
and i18=NULL
and base_view_name='test41'
```



#### 4.3.1.2 Execute as a SELECT statement in the VQL Shell using a template

In order to obtain the VQL statements you have to specify the `datasource_name` and the `base_view_name`. And also choose the database where the base view would be stored, and a folder if it is the case.

If you use another data source as a template in order to copy the route configuration you will need to introduce the name of the database (`template_database`) and the name of the template data source (`template_datasource`). This way parameters `route_type`, `authentication`, `username`, `password` and `encrypted_password` will be

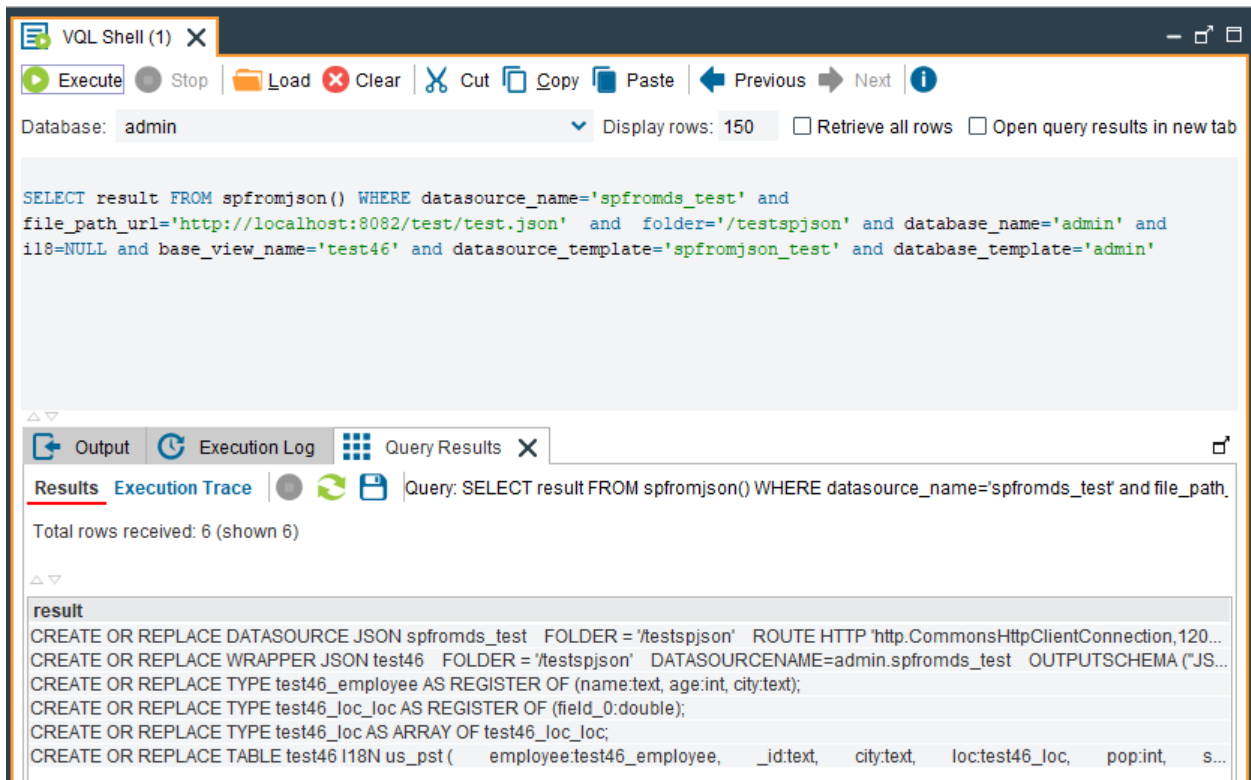
imported from the template. But you have to introduce the `file_path_url` parameter, which is mandatory as the file path of the template data source will be overwritten.

The `i18` parameter is also mandatory.

#### 4.3.1.2.1 Sample and results

```
SELECT result FROM GENERATE_VQL_TO_CREATE_JSON_DATA_SOURCE_BASE_VIEW()
WHERE datasource_name = 'gen_test_json'
and file_path_url = 'http://localhost:8082/test/test.json'
and folder = '/testspjson'
and database_name = 'admin'
and i18 = NULL
and base_view_name = 'test46'
and template_datasource = 'spfromjson_test'
and template_database = 'admin'
```

## 4.4



The screenshot shows the VQL Shell interface. The query entered is:

```
SELECT result FROM spfromjson() WHERE datasource_name='spfromds_test' and
file_path_url='http://localhost:8082/test/test.json' and folder='/testspjson' and database_name='admin' and
i18=NULL and base_view_name='test46' and datasource_template='spfromjson_test' and database_template='admin'
```

The results pane shows the following output:

```
CREATE OR REPLACE DATASOURCE JSON spfromds_test FOLDER = '/testspjson' ROUTE HTTP 'http.CommonsHttpClientConnection,120...
CREATE OR REPLACE WRAPPER JSON test46 FOLDER = '/testspjson' DATASOURCENAME=admin.spfromds_test OUTPUTSCHEMA ("JS...
CREATE OR REPLACE TYPE test46_employee AS REGISTER OF (name:text, age:int, city:text);
CREATE OR REPLACE TYPE test46_loc_loc AS REGISTER OF (field_0:double);
CREATE OR REPLACE TYPE test46_loc AS ARRAY OF test46_loc_loc;
CREATE OR REPLACE TABLE test46 I18N us_pst ( employee:test46_employee, _id:text, city:text, loc:test46_loc, pop:int, s...
```

## 4.5 GENERATE\_VQL\_TO\_CREATE\_JSON\_DATA\_SOURCE\_BASE\_VIEW

### 4.5.1 Description

The `GENERATE_VQL_TO_CREATE_JSON_DATA_SOURCE_BASE_VIEW` stored procedure returns the VQL statements required to create a JSON data source and a JSON base view from a JSON file. With this stored procedure if the `route_type` is HTTP, using `relative_url` parameter, we can differentiate the data source URL and base view URL, the base URL(data source) will be concatenated to base view's relative URL. Note that it does not actually create the view, it only returns the VQL statements to do so. This stored procedure requires VDP 8.0 update 20220815 or later.

### 4.5.2 Syntax

## 4.6

```
GENERATE_VQL_TO_CREATE_JSON_DATA_SOURCE_BASE_VIEW (  
  data_source_name : text,  
  route_type: text,  
  file_path_url: text,  
  relative_url: text,  
  authentication: text,  
  username: text,  
  password: text,  
  encrypted_password: text,  
  folder : text,  
  database_name : text,  
  i18n : text,  
  aws_access_key : text,  
  aws_secret_key : text,  
  aws_encrypted_secret_key : boolean,  
  aws_iam_role_arn : text,  
  use_aws_instance_credentials : boolean,  
  aws_route_pattern : text,  
  aws_custom_properties : text,  
  base_view_name : text,  
  base_view_folder : text  
)
```

- ***data\_source\_name***: name of the data source. Mandatory.
- ***database\_name***: name of the database in which artifacts will be created. If null, the procedure uses the current database.
- ***base\_view\_name***: name of the base view to be created. If null, the name will be auto-generated.
- ***route\_type***: specify the types of paths to access the file/s. Mandatory no template is specified.
  - LOCAL: the file or set of files to be used are located in the server machine.

- HTTP: the data to be used has to be obtained by sending an HTTP request to a server.
- **file\_path\_url**: route to the delimited file or files. It can be a file path or a url. If the route\_type is http ,it is data source's URL, it can be base url or absolute url including the filename. Mandatory.
- **relative\_url**: Only it is used when the route\_type is HTTP, and it is optional. Base view's relative URL (for example: "/books.json"), this URL will be concatenated to the Base URL when executing the base view (the resulting URL will be http://www.test.com/books.json).
- **authentication**: authentication type (OFF|BASIC). Mandatory no template is specified.
- **username**: HTTP user.
- **password**: HTTP password.
- **encrypted\_password**: whether the password has been specified encrypted (as per Denodo VQL).
- **folder**: folder in which to place the data source, if the base view folder is not specified, the base view will be stored in the same folder. The result will include the VQL statements to create this folder(s). If null, the VQL will not specify a folder.
- **i18n**: i18n of the base view to be created. If null, the i18n of the current database will be assigned.
- **aws\_access\_key**: the AWS access key ID.
- **aws\_secret\_key**: the AWS secret key.
- **aws\_encrypted\_secret\_key**: whether the AWS secret key has been specified encrypted (as per Denodo VQL).
- **aws\_iam\_role\_arn**: Optionally, you can specify an AWS IAM role ARN to get the necessary privileges to connect to the database.
- **use\_aws\_instance\_credentials**: automatically obtain the credentials of the database from the AWS instance where this Virtual DataPort server is running.
- **aws\_route\_pattern**: Only for delimited-file data sources: if the path points to a directory and you enter a value in File name pattern, the data source will only process the files whose name matches the regular expression entered in this box. For example, if you only want to process the files with the extension log, enter (\*.\*)\log.
- **aws\_custom\_properties**: set the same properties that you would put in the Hadoop configuration files like core-site.xml to configure the S3A Hadoop connector.
- **base\_view\_folder**: folder in which to place the base view. The result will include the VQL statements to create this folder(s). If null, the base view will be stored in the same folder as the datasource.



### 4.6.1 Execution

In order to obtain the VQL statements you have to specify the `datasource_name` and the `base_view_name`. And also choose the database where the base view would be stored, and a folder if it is the case.

All the parameters related with the datasource have to be introduced: `route_type`, `authentication` and `file_path_url`. Authentication parameters `username`, `password` and `encrypted_password` may be necessary also.

Belonging to the base view there is the optional `i18n` parameter. And if the data source has a base URL, `relative_url` parameter allows having a relative URL in the base view.

#### 4.6.1.1 Sample

```
SELECT result FROM GENERATE_VQL_TO_CREATE_JSON_DATA_SOURCE_BASE_VIEW()  
WHERE datasource_name='spfromds_test_2'  
and route_type='HTTP'  
and file_path_url='https://filesamples.com/samples/code/json'  
and authentication='OFF'  
and folder='/testspjson'  
and database_name='admin'  
and base_view_name='spfromds_testbv_2'  
and relative_url= 'sample4.json'
```

## 4.7 GENERATE\_VQL\_TO\_CREATE\_JSON\_DATA\_SOURCE

### 4.7.1 Description

The `GENERATE_VQL_TO_CREATE_JSON_DATA_SOURCE` stored procedure returns the VQL statements required to create a JSON data source.. Note that it does not actually create the view, it only returns the VQL statements to do so. This stored procedure requires VDP 8.0 update 20220815 or later.

### 4.7.2 Syntax

## 4.8

```
GENERATE_VQL_TO_CREATE_JSON_DATA_SOURCE (  
  data_source_name : text,  
  database_name : text,  
  route_type: text,  
  file_path_url: text,  
  authentication: text,  
  username: text,
```

```
password: text,
encrypted_password: text,,
aws_access_key : text,
aws_secret_key : text,
aws_encrypted_secret_key : boolean,
aws_iam_role_arn : text,
use_aws_instance_credentials : boolean,
aws_route_pattern : text,
aws_custom_properties : text,
folder : text )
```

- **data\_source\_name**: name of the data source. Mandatory.
- **database\_name**: name of the database in which artifacts will be created. If null, the procedure uses the current database.
- **route\_type**: specify the types of paths to access the file/s. Mandatory no template is specified.
  - LOCAL: the file or set of files to be used are located in the server machine.
  - HTTP: the data to be used has to be obtained by sending an HTTP request to a server.
  - S3: Use this type of route to obtain the data from a file or a set of files located in a S3 bucket.
- **file\_path\_url**: route to the delimited file or files. It can be a file path or a url. If the route\_type is http ,it is data source's URL, it can be base url or absolute url including the filename. Mandatory.
- **authentication**: authentication type (OFF|BASIC). Mandatory no template is specified.
- **username**: HTTP user.
- **password**: HTTP password.
- **encrypted\_password**: whether the password has been specified encrypted (as per Denodo VQL).
- **aws\_access\_key**: the AWS access key ID.
- **aws\_secret\_key**: the AWS secret key.
- **aws\_encrypted\_secret\_key**: whether the AWS secret key has been specified encrypted (as per Denodo VQL).
- **aws\_iam\_role\_arn**: Optionally, you can specify an AWS IAM role ARN to get the necessary privileges to connect to the database.
- **use\_aws\_instance\_credentials**: automatically obtain the credentials of the database from the AWS instance where this Virtual DataPort server is running.
- **aws\_route\_pattern**: Only for delimited-file data sources: if the path points to a directory and you enter a value in File name pattern, the data source will only process the files whose name matches the regular expression entered in this box. For example, if you only want to process the files with the extension log, enter (\*.\*)\log.

- ***aws\_custom\_properties***: set the same properties that you would put in the Hadoop configuration files like core-site.xml to configure the S3A Hadoop connector.
- ***folder***: folder in which to place the created structures. The result will include the VQL statements to create this folder(s). If null, the VQL will not specify a folder.

### 4.8.1 Execution

In order to obtain the VQL statements you have to specify the `datasource_name` And also choose the database where the base view would be stored, and a folder if it is the case.

All the parameters related with the datasource have to be introduced: `route_type`, `authentication` and `file_path_url`. Authentication parameters `username`, `password` and `encrypted_password` may be necessary also.

#### 4.8.1.1 Sample

```
SELECT result FROM GENERATE_VQL_TO_CREATE_JSON_DATA_SOURCE()  
WHERE datasource_name='spfromds_test'  
and route_type='HTTP'  
and file_path_url='https://filesamples.com/samples/code/json'  
and authentication='OFF'  
and folder='/testspjson'  
and database_name='admin'
```

## 4.9 GENERATE\_VQL\_TO\_CREATE\_JSON\_BASE\_VIEW

### 4.9.1 Description

The `GENERATE_VQL_TO_CREATE_JSON_BASE_VIEW` stored procedure returns the VQL statements required to create a JSON base view from a JSON file. If the route type of the data source is HTTP, and the path is not a complete URL, the `relative_url` would be added to the base URL. Note that it does not actually create the view, it only returns the VQL statements to do so. This stored procedure requires VDP 8.0 update 20220815 or later.

## 4.9.2 Syntax

### 4.10

```
GENERATE_VQL_TO_CREATE_JSON_BASE_VIEW (  
  data_source_name : text,  
  relative_url: text,  
  database_name : text,  
  i18n : text,  
  base_view_name : text,  
  base_view_folder : text  
  
)
```

- **data\_source\_name**: name of the data source. Mandatory.
- **database\_name**: name of the database in which artifacts will be created. If null, the procedure uses the current database.
- **base\_view\_name**: name of the base view to be created. If null, the name will be auto-generated.
- **relative\_url**: Only it is used when the route\_type is HTTP, and it is optional. Base view's relative URL (for example: "/books.json"), this URL will be concatenated to the Base URL when executing the base view (the resulting URL will be http://www.test.com/books.json).
- **i18n**: i18n of the base view to be created. If null, the i18n of the current database will be assigned
- **base\_view\_folder**: folder in which to place the base view. The result will include the VQL statements to create this folder(s). If null, the base view will be stored in the same folder as the datasource.

#### 4.10.1 Execution

In order to obtain the VQL statements you have to specify the `datasource_name` and the `base_view_name`. And also choose the database where the dat source is stored. In addition there is the optional `i18n` parameter.

##### 4.10.1.1 Sample

```
SELECT result FROM GENERATE_VQL_TO_CREATE_JSON_BASE_VIEW()  
WHERE datasource_name='spfromds_test'  
and database_name='admin'  
and base_view_name='spfromds_testbv'
```

```
and relative_url= 'sample4.json'
```

## 5 EXCEL

---

### 5.1 INSTALLATION

#### 5.1.1 Importing the Stored Procedure

In order to run the `GenerateVQLForExcelFullDataSourceBaseView` stored procedure you have to load the `denodo-vqlgeneration.jar` file, using the `File > Jar Management` option of the VDP Administration Tool.

The class `GenerateVQLToCreateExcelBaseViewStoredProcedure` is deprecated.

#### 5.1.2 Adding the Stored Procedure

##### 5.1.2.1 VQL Shell

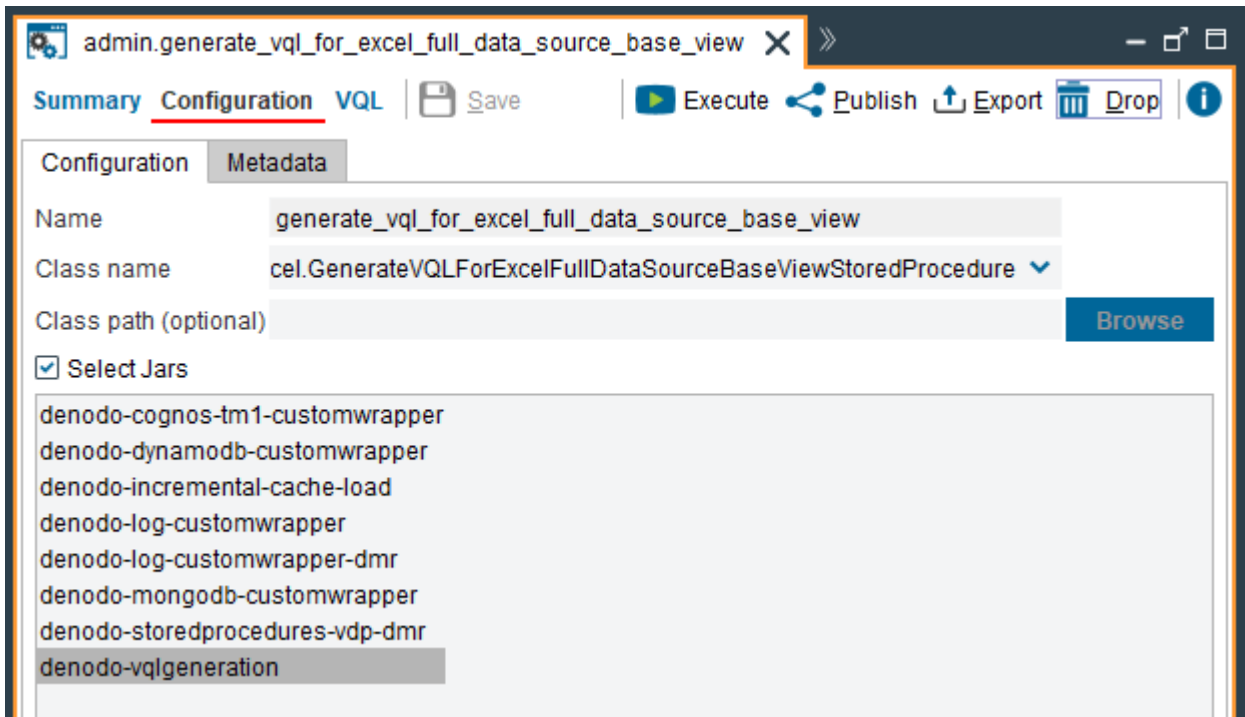
You can add the stored procedure programmatically with a `CREATE PROCEDURE` statement:

```
CREATE [OR REPLACE] PROCEDURE <name:identifier>  
CLASSNAME='com.denodo.connect.storedprocedure.excel.GenerateVQLForExcelFull  
DataSourceBaseView'  
JARS 'denodo-vqlgeneration';  
  
[ FOLDER = <literal> ]  
[ DESCRIPTION = <literal> ]
```

##### 5.1.2.2 Virtual DataPort Administration Tool menu

You can also add the stored procedure by clicking on `Stored procedure` on the menu `File > New`:

You must input a name in the `Name` field and select the `Select Jars` checkbox in order to use the jar file, `denodo-vqlgeneration`, previously added to the Virtual DataPort Server.



## 5.2 DESCRIPTION

The `GENERATE_VQL_FOR_EXCEL_FULL_DATA_SOURCE_BASE_VIEW` stored procedure returns the VQL statements required to create a Excel base view from a Excel file. The stored procedure allows you to use an existing data source as a template to copy the configuration of the route, minus the field path or the url. Note that it does not actually create the view, it only returns the VQL statements to do so.

## 5.3 SYNTAX

```
GENERATE_VQL_FOR_EXCEL_FULL_DATA_SOURCE_BASE_VIEW (
  data_source_name : text,
  route_type: text,
  file_path_url: text,
  authentication: text,
  username: text,
  password: text,
  encrypted_password: text,
  folder : text,
  database_name : text,
  i18n : text,

  base_view_name : text,
  base_view_folder : text,
  aws_access_key : text,
  aws_secret_key : text,
  aws_encrypted_secret_key : boolean,
  aws_iam_role_arn : text,
```

```
use_aws_instance_credentials : boolean,  
aws_route_pattern : text,  
aws_custom_properties : text,  
template_datasource: text,  
template_database: text,  
type_of_file:text,  
worksheets:text,  
extract_data_from_all_sheets:boolean,  
extract_sheet_name_as_a_new_column:boolean,  
start_cell:text,  
end_cell:text,  
has_headers:boolean,  
stream_tuples:boolean  
)
```

- **data\_source\_name**: name of the data source. Mandatory.
- **database\_name**: name of the database in which artifacts will be created. If null, the procedure uses the current database.
- **base\_view\_name**: name of the base view to be created. If null, the name will be auto-generated.
- **route\_type**: specify the types of paths to access the file/s. Mandatory no template is specified.
  - LOCAL: the file or set of files to be used are located in the server machine.
  - HTTP: the data to be used has to be obtained by sending an HTTP request to a server.
  - S3: Use this type of route to obtain the data from a file or a set of files located in a S3 bucket.
- **file\_path\_url**: route to the delimited file or files. It can be a file path or a url. Mandatory.
- **authentication**: authentication type (OFF|BASIC). Mandatory no template is specified.
- **username**: HTTP user.
- **password**: HTTP password.
- **encrypted\_password**: whether the password has been specified encrypted (as per Denodo VQL).
- **folder**: folder in which to place the data source, if the base view folder is not specified, the base view will be stored in the same folder. The result will include the VQL statements to create this folder(s). If null, the VQL will not specify a folder.
- **i18n**: i18n of the base view to be created. If null, the i18n of the current database will be assigned.
- **base\_view\_folder**: folder in which to place the base view. The result will include the VQL statements to create this folder(s). If null, the base view will be stored in the same folder as the datasource.
- **aws\_access\_key**: the AWS access key ID.
- **aws\_secret\_key**: the AWS secret key.



- ***aws\_encrypted\_secret\_key***: whether the AWS secret key has been specified encrypted (as per Denodo VQL).
- ***aws\_iam\_role\_arn***: Optionally, you can specify an AWS IAM role ARN to get the necessary privileges to connect to the database.
- ***use\_aws\_instance\_credentials***: automatically obtain the credentials of the database from the AWS instance where this Virtual DataPort server is running.
- ***aws\_route\_pattern***: Only for delimited-file data sources: if the path points to a directory and you enter a value in File name pattern, the data source will only process the files whose name matches the regular expression entered in this box. For example, if you only want to process the files with the extension log, enter `(.*)\.log`.
- ***aws\_custom\_properties***: set the same properties that you would put in the Hadoop configuration files like `core-site.xml` to configure the S3A Hadoop connector.
- ***template\_database***: name of the database where the template data source lives. If null and a template data source is specified, the current database will be assumed to contain it.
- ***template\_datasource***: name of the data source from which the route configuration will be copied in the VQL. This parameter is mutually exclusive with `authentication`, `username`, `password` and `encrypted_password`. The route path will be modified by the field `file_path_url`.
- ***type\_of\_file***: Depending on the version of the file:
 

"Excel	97-2003	"(*.xls)"
"Excel 2007 or later (*.xlsx)"		

The option "Stream tuples" will be enabled or not (see below).
- ***worksheets***: (optional). Name of the worksheet that you want to extract data from.  
If left blank, the first sheet of the file is used by default.
- ***extract\_data\_from\_all\_sheets***: if true, the base view will retrieve data from all sheets,
- ***extract\_sheet\_name\_as\_a\_new\_column***: boolean, if you want the base view to have an extra column that will contain the name of the worksheet from which each row is obtained from. The default name of this extra column is "sheet\_name".
- ***start\_cell, end\_cell***: (optional): first and last cell of the sheet that will be processed.  
If "StartCell" is B2 and "EndCell" is Z99, the view will return the data in the range B2:Z99.
- ***has\_headers***: If true, it is assumed that the first tuple extracted from the file data area contains the names of the fields. These names will be used to create the attributes of the base relation for Virtual DataPort.
- ***stream\_tuples***: if selected, the file will be read using a SAX parser (only available for Excel 2007 files (\*.xlsx)). If cleared, it will be read using a DOM parser.

## 5.4 EXECUTION

### 5.4.1 Execute as a SELECT statement in the VQL Shell without template

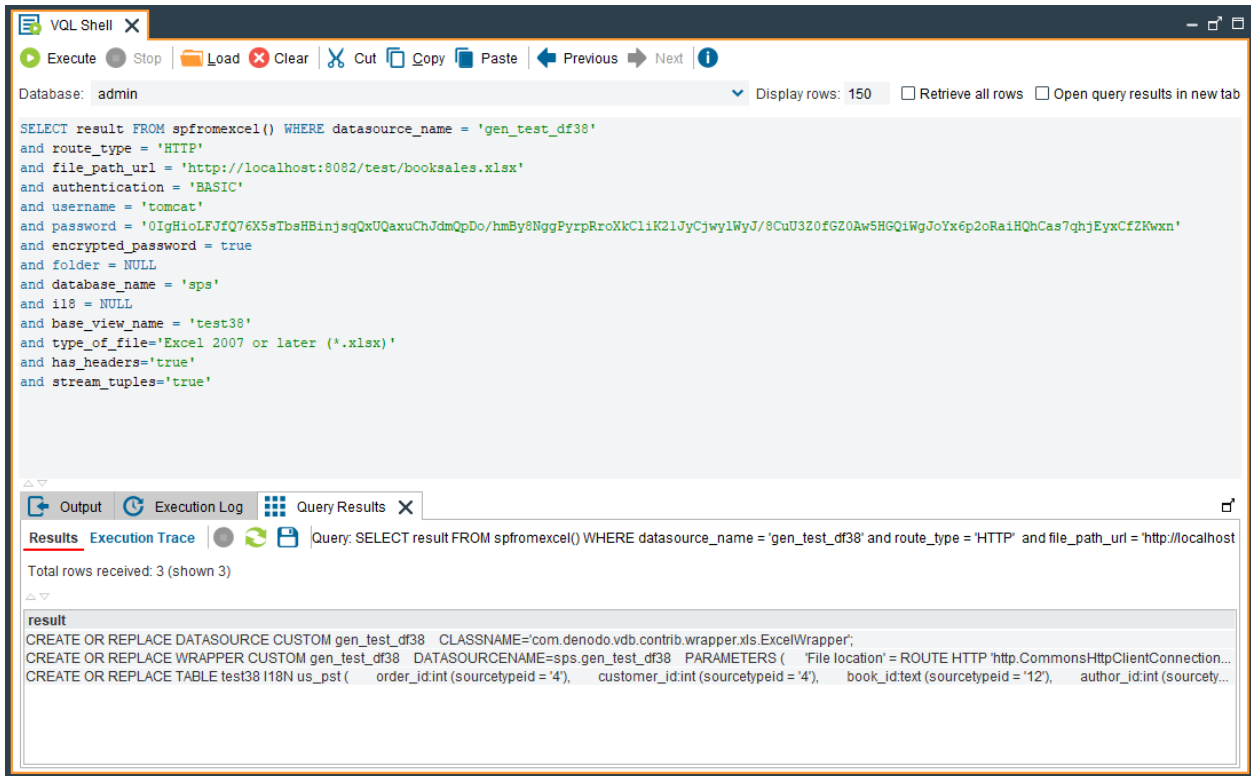
In order to obtain the VQL statements you have to specify the `datasource_name` and the `base_view_name`. And also choose the database where the base view would be stored, and a folder if it is the case.

All the parameters related with the datasource have to be introduced: `route_type`, `authentication` and `file_path_url`. Authentication parameters `username`, `password` and `encrypted_password` may be necessary also.

In addition, Excel base views require the following parameter: `type_of_file`. And also other optional parameters: `worksheets`, `extract_data_from_all_sheets`, `extract_sheet_name_as_a_new_column`, `ignore_matching_errors`, `start_cell`, `end_cell`, `has_headers` and `stream_tuples`.

#### 5.4.1.1 Sample and results

```
SELECT result FROM GENERATE_VQL_FOR_EXCEL_FULL_DATA_SOURCE_BASE_VIEW()
WHERE datasource_name = 'gen_test_excel38'
and route_type = 'HTTP'
and file_path_url = 'http://localhost:8082/test/booksales.xlsx'
and authentication = 'BASIC'
and username = 'tomcat'
and password =
'0IghioLFJfQ76X5sTbsHBinjsqQxUQaxuChJdmQpDo/hmBy8NggPyrpRroXkCliK21JyCjwylW
yJ/8CuU3Z0fGZ0Aw5HGQiWgJoYx6p2oRaiHQhCas7qhjEyxCfZKwxn'
and encrypted_password = true
and folder = NULL
and database_name = 'sps'
and i18 = NULL
and base_view_name = 'test38'
and type_of_file='Excel 2007 or later (*.xlsx)'
and has_headers='true'
and stream_tuples='true'
```



### 5.4.2 Execute as a SELECT statement in the VQL Shell using a template

In order to obtain the VQL statements you have to specify the `datasource_name` and the `base_view_name`. And also choose the database where the base view would be stored, and a folder if it is the case.

If you use another data source as a template in order to copy the route configuration you will need to introduce the name of the database (`template_database`) and the name of the template data source (`template_datasource`). This way parameters `route_type`, `authentication`, `username`, `password` and `encrypted_password` will be imported from the template. But you have to introduce the `file_path_url` parameter, which is mandatory as the file path of the template data source will be overwritten. Note that despite using the template, parameters corresponding with the base view (and not the data source) can be used: `type_of_file`(mandatory), `worksheets`, `extract_data_from_all_sheets`, `extract_sheet_name_as_a_new_column`, `ignore_matching_errors`, `start_cell`, `end_cell`, `has_headers` and `stream_tuples`.

#### 5.4.2.1 Sample and results

```
SELECT result FROM GENERATE_VQL_FOR_EXCEL_FULL_DATA_SOURCE_BASE_VIEW()  
WHERE  
datasource_name = 'gen_test_template_excel38'  
and file_path_url = 'http://localhost:8082/test/users.xlsx'  
and template_datasource='gen_test_excel38'  
and template_database='sps'  
and folder = NULL  
and database_name = 'sps'  
and i18 = NULL  
and base_view_name = 'exceltemplate38'  
and type_of_file='Excel 2007 or later (*.xlsx)'  
and has_headers='false'  
and stream_tuples='false'
```

```
CREATE OR REPLACE TABLE exceltemplate38 FOR US_PST ( column_0.double (sourcetype = 0), column_1.text (sourcetype = 12), column_2.text (sourcetype = 12), column...
```

## 6 APPENDICES

---

### 6.1.1 Appendix I. Example of S3 Route execution.

```
SELECT result FROM dffull() WHERE datasource_name='spfromds_testdf' and
route_type='S3'
and file_path_url='s3://denodo-labs-dfs/testvql/Schools.csv'
and authentication='OFF'
and base_view_name='s3_df'
and base_view_folder='/testbv'
and aws_access_key='AKIAUJBRL5LBHQMAAAG7'
and
aws_secret_key=
'MpZpvt6KwAKArX*****Hk+n295Pt9oZ5ENCGF3B2QqcAbdjqrD1lsah0+pWkD8YCyU
U4Rqw2wfxCJVbR1p1OM6VEwElJP1wFQbpmb1VZt6j505FPCxUBbjpt04rX2LewuC7QA='
and aws_encrypted_secret_key=true
and aws_iam_role_arn= 'arn:aws:iam::678559877122:role/DenodoLabsTestRole'
and database_name='vqlgeneration'
and i18=NULL and column_delimiter=', '
and eol_delimiter= '\n' and has_headers = TRUE and
ignore_matching_errors=true
```

## 6.1.2

### 6.1.3 Results

The screenshot shows a VQL Shell window with the following components:

- Toolbar:** Execute, Stop, Load, Clear, Cut, Copy, Paste, Previous, Next, and an information icon.
- Database:** vqlgeneration
- Limit displayed rows:** 150
- Options:** Retrieve all rows (unchecked), Open query results in new tab (unchecked).
- Query:**

```
SELECT result FROM dffull() WHERE datasource_name='spfromds_testdf2' and route_type='S3'
and file_path_url='s3://denodo-labs-dfs/testvql/Schools.csv'
and authentication='OFF'
and base_view_name='s3_df'
and base_view_folder='/testbv'
and aws_access_key='AKIAUJBRL5LBHQMAAAG7'
and aws_secret_key= 'MpZpvt6KwAKArXI+zmOV6TRDLzK5oCFOQjYrByb5X3VXT+VksUAREPB8YHk+n295Pt9oZ5ENCGF3B2QqcAbdjqrD1lsah
bjpt04rX2LewuC7QA='
and aws_encrypted_secret_key=true
and aws_iam_role_arn='arn:aws:iam::678559317122:role/DenodoLabsTestRole'
and database_name='vqlgeneration'
and i18=NULL and column_delimiter=', '
and eol_delimiter= '\n' and has_headers = TRUE and ignore_matching_errors=true
```
- Output Panel:**
  - Buttons: Output, Execution Log, Query Results.
  - Query: SELECT result FROM dffull() WHERE datasource\_name='spfromds\_testdf2' and route\_type='S3'
  - Total rows received: 3 (shown 3)
  - Result:
 

```
CREATE OR REPLACE DATASOURCE DF spfromds_testdf2 ROUTE S3 'hdfs.S3Connection' 's3://denodo-labs-dfs/testvql/Schools.csv' AU...
CREATE OR REPLACE WRAPPER DF s3_df FOLDER = 'testbv' DATASOURCENAME=vqlgeneration.spfromds_testdf2 OUTPUTSCHEMA...
CREATE OR REPLACE TABLE s3_df I18N es_euro ( schoolid:text, name_full:text, city:text, state:text, country:text ) FOLDE...
```