# **Extension Data Sources:**

We have provided extension datasource support to create or view reports with custom datasource types like SSAS, OData, MySQL, etc. We have to do some manual changes in order to create and run the reports. The below describes procedure to work with custom data extension.

# Create extension and modifying configuration.

We must add the new extension to configuration file and build extension assemblies with Syncfusion assemblies to add new extension type. Here we are going to give demo of creating custom data extension for MySql datasource type.

## Creating extension assembly

1. Create class library using visual studio and name the project as “Syncfusion.Reporting.DataExtension.MySql”.
2. We need to compile two extension project one is for reportserver and another for reportdesigner. Add the reference “Syncfusion.ReportControls.Wpf” for the extension project that we going to build for report designer.

**Note: We need to add Syncfusion.Ej.ReportViewer assemblies as reference for building extension for reportserver.**

1. Add a class “MySqlDataExtension” and inherit the “IDataExtension” and implement the interface methods.
2. We need to add the codes to “GetData” method for retrieving datas from the connection and return data table as like below.

|  |
| --- |
| public object GetData()  {  var connectionString = this.GetConnectionString();  using (MySqlConnection connection = new MySqlConnection(connectionString))  {  try  {  connection.Open();  DataTable dataTable = this.GetTable(connection, this.CommandText);  return dataTable;  }  catch (Exception ex)  {  throw ex;  }  finally  {  connection.Close();  connection.Dispose();  }  }  } |

1. The “ValidateConnection” method is used to test the connection and it will be raised from source when click on test connection option in reportserver and report designer. The below is the sample code for MySql.

|  |
| --- |
| public bool ValidateConnection(string connectionString)  {  MySqlConnection connection = new MySqlConnection(connectionString);  try  {  connection.Open();  return true;  }  catch  {  return false;  }  finally  {  connection.Close();  connection.Dispose();  }  } |

1. After completing the changes build the project.

## Modifying the configuration files

1. Create a configuration file and name it as “DataExtension.config”.
2. The following “configSections” section is mandatory to process the extension assemblies inside source, so add it as same as given in the below.

|  |
| --- |
| <configSections>  <section name="ReportingExtensions" type="Syncfusion.Reporting.Extensions, Syncfusion.ReportControls.WPF" allowLocation="true" allowDefinition="Everywhere" />  </configSections> |

**Note:** This “configSections” not necessary for reportserver.

1. We have to add the tag “Extension” for all the newly added extension types. It has following attributes.

|  |  |
| --- | --- |
| **Attribute Name** | **Description** |
| Name | Name of your data source type that going to display in list. |
| Assembly | Name of newly created extension assembly. |
| Type | Extension class name with the namespace. |

In the below code sample we have given MySql data extension as demo.

|  |
| --- |
| <Extension Name="MySql" Assembly="Syncfusion.Reporting.DataExtensions.MySql" Type="Syncfusion.Reporting.DataExtensions.MySql.MySqlDataExtension"/>  </DataExtension> |

1. The resultant configuration will be as like below.

|  |
| --- |
| <configuration>  <configSections>  <section name="ReportingExtensions" type="Syncfusion.Reporting.Extensions, Syncfusion.ReportControls.WPF" allowLocation="true" allowDefinition="Everywhere" />  </configSections>  <ReportingExtensions>  <DataExtension>  <Extension Name="MySql" Assembly="Syncfusion.Reporting.DataExtensions.MySql" Type="Syncfusion.Reporting.DataExtensions.MySql.MySqlDataExtension"/>  </DataExtension>  </ReportingExtensions>  </configuration> |

# Replacing the assemblies and editing the data extension

## Report Designer

For report designer we need to replace the newly created assembly’s and its depended assembly in the following location.

C:\Program Files (x86)\Syncfusion\Report Designer\ReportDesigner

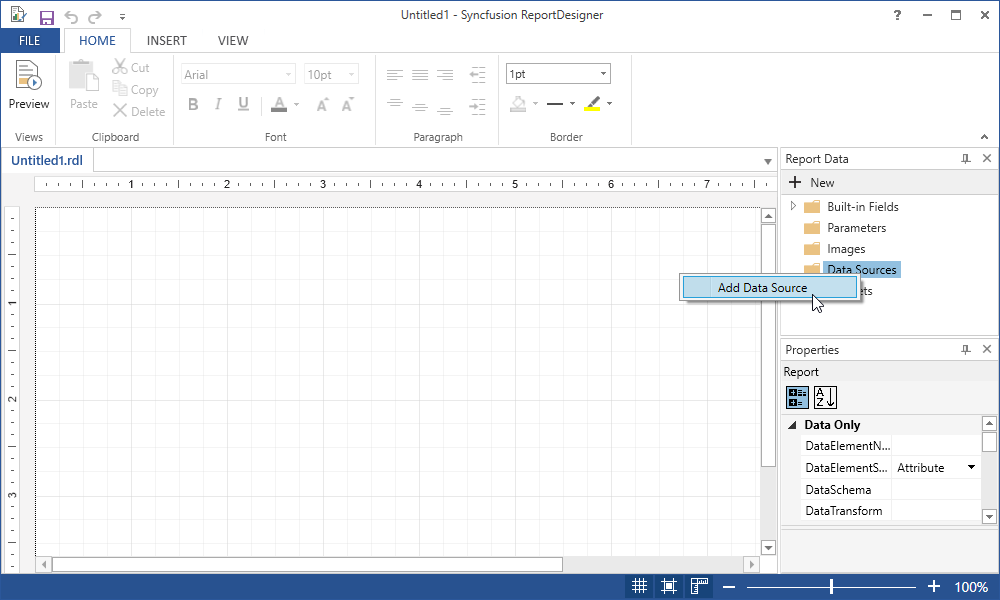
Also need to add the created/modified “DataExtension.config” file above location.

|  |
| --- |
| <ReportingExtensions>  <DataExtension>  <Extension Name="MySql" Assembly="Syncfusion.Reporting.DataExtensions.MySql" Type="Syncfusion.Reporting.DataExtensions.MySql.MySqlDataExtension"/>  </DataExtension>  </ReportingExtensions> |

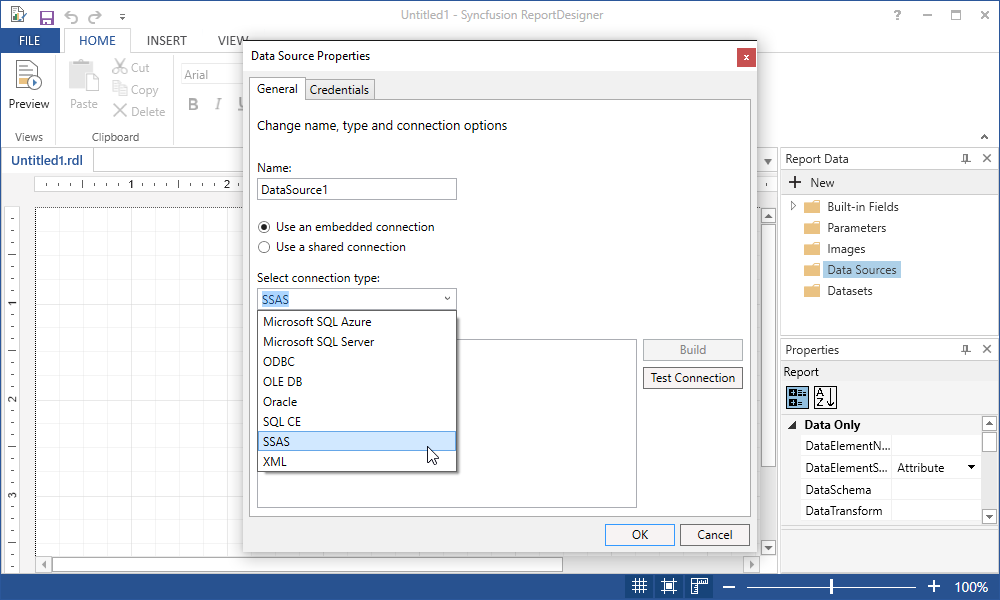
# Running the report designer

The following steps shows the procedure to test SSAS datasource in Report designer after completion of the configuration and assembly changes.

1. Run report designer utility.
2. Right Click on DataSources in Report Data to add datasource.

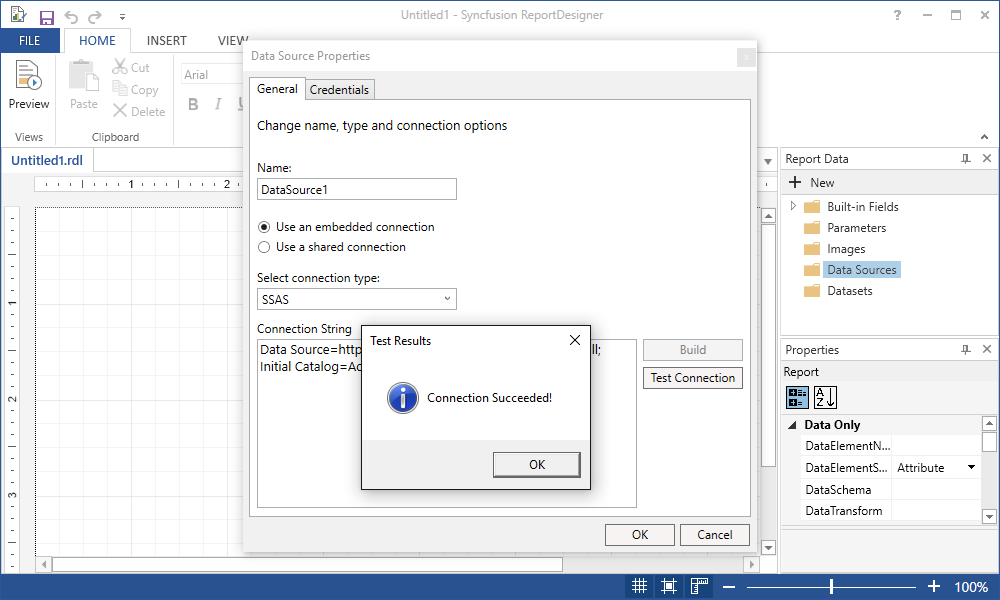


1. Now, in datasource properties dialog you can view added external datasource under connection type.

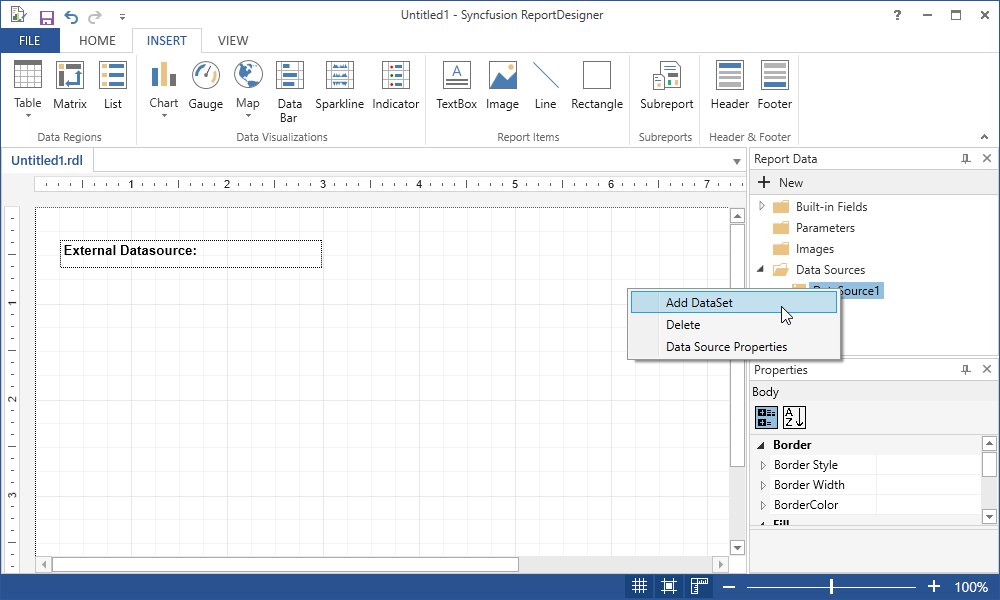


1. Give connection string to connect with external datasource.

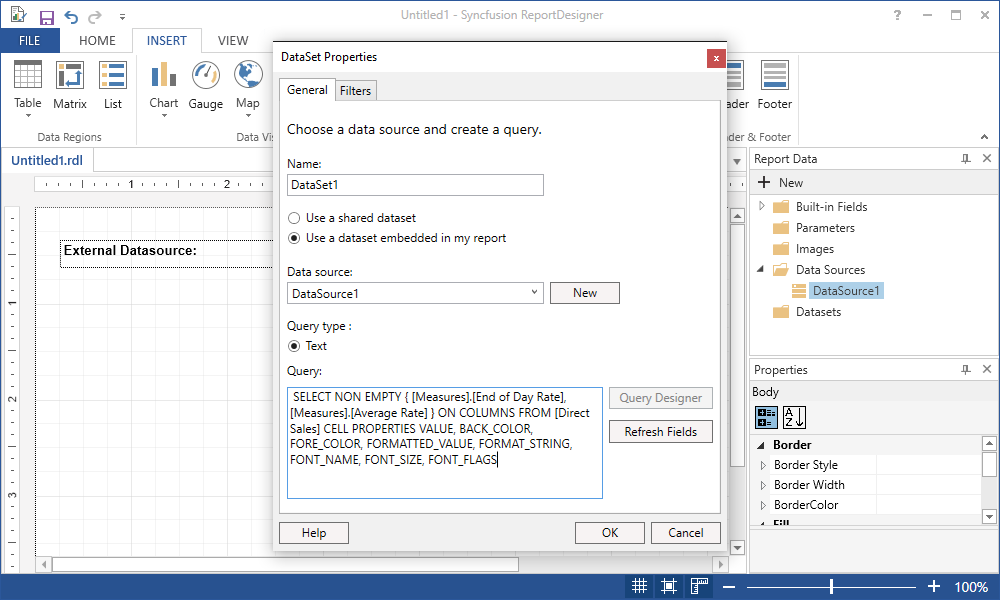
|  |
| --- |
| Data Source=http://bi.syncfusion.com/olap/msmdpump.dll; Initial Catalog=Adventure Works DW 2008 SE; |

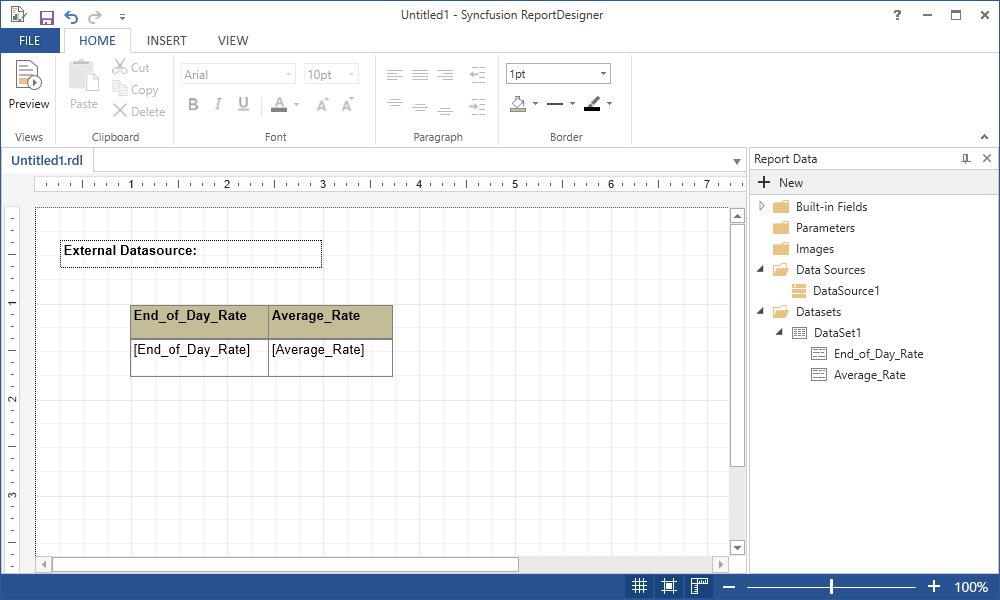


1. You can add dataset to external datasources,



1. Provide the query command text in dataset properties dialog.





# Samples:

The following table has the location of project samples need to build a custom data extension.

|  |  |
| --- | --- |
| **Sample** | **Download link** |
| Data extension project for report designer | <http://www.syncfusion.com/downloads/support/directtrac/general/ze/ReportDesignerDataExtension2053795980> |