

# Applenti - Report Designer

In this documentation we have provided the steps to follow to set up the Report designer

Our Report Designer offers a complete framework for producing daily reports from any database or any No SQL source.

The product focuses on easy installation and report design: Once setup, reports can be built and published in a minute.

## Main Features

- **Dynamic SQL sources:** Use either your SQL or let the Seal engine build dynamically the SQL used to query your database.
- **LINQ queries:** Join and query any Data Sources (SQL, Excel, XML, Olap Cube, HTTP JSON, etc.) with the power of LINQ queries.
- **Native Pivot Tables:** Simply drag and drop elements directly in a pivot table (Cross Tab) and display them in your report.
- **HTML 5 Charts:** Define and display Chart Series in two mouse clicks (Support of ChartJS, NVD3 and Plotly libraries).
- **Fully Responsive and HTML Rendering using Razor engine:** Use the power of HTML5 in the report result (Bootstrap layout, Responsiveness, Tables sorting and filtering). Customize your report presentation in HTML with the Razor engine parsing.
- **Web Report Server:** Publish and edit your reports on the web (Support of Windows and Linux OS with .NET Core).
- **Drill Down navigation and Sub Reports:** Navigate in your report result to drill to a detail or to execute another report.
- **Low TCO (total cost of ownership):** The product is designed for minimal ongoing maintenance

## Main Entities: Repository and Reports

The **Seal Repository** stores all the entities managed by Seal Report: Data Sources, Devices, and Reports.

In addition, the repository contains extra folders for the Settings (configuration and dictionary for translations), the Views (used to render report result), and the Security (configuration and providers).

By default, the repository root folder is located in "*C:\ProgramData\Seal Report Repository*".

A report is composed of Data Sources, Models, and Views. The report may also reference Views Templates located in the repository.

- **Data Sources** contain the descriptions of database connections, tables, joins and columns.
- **Models** define how to generate the Result Set (Data Table) and Series from a single SQL Statement.
- The **Views** are used to generate a HTML document from the **Models** using Razor parsing and rendering.
- **Tasks** may be defined to perform SQL or Script tasks.

## Product Components

Seal Report is composed of the **Server Manager** application, the **Report Designer** application, the **Web Report Server**, the **Task Scheduler** and the **Seal Report Scheduler**.

- The **Server Manager** (a Windows application) edits repository Data Sources, Devices and is used to perform administrative tasks (checking sources and reports, publishing web site, configuring the server, editing the security, etc.).
- The **Report Designer** (a Windows application) creates, edits and executes reports.

## Repository Folders

Several sub-folders are located from the Repository Root folder. These folders contain all files necessary to run Seal Report.

- **Databases:** dedicated folder for local database files (e.g. an MS Access file, an Excel file).
- **Reports:** the reports (\*.srex), files and sub-folders published by the Web Report Server. The Reports root folder is the parent of all folders, files and reports published.
- **Settings:** the Server configuration (Configuration.xml) and the translation files (\*.csv).
- **Sources:** the data source files (\*.scfx) available for the reports. These data sources are shared amongst all the reports.
- **Sources\TableTemplates:** table templates (\*.cshtml) referenced by No SQL tables defined in LINQ Data Sources.
- **SubReports:** the sub-reports (\*.srex) referenced by an element in a data source. Sub-Reports allow master-detail navigation from a report result.
- **Views:** the view templates (\*.cshtml) used by the report views with their configuration and their related JavaScript, CSS and image files.

## Setup

Run the executable file (provided under releases in the Github link-

[https://github.com/nehanandank/applenti\\_sealReport](https://github.com/nehanandank/applenti_sealReport))

The setup will install the Report Designer, the Server Manager with a default repository.

All executables, libraries and configuration files are installed by default in C:\Program Files\Seal Report.

All repository files are installed by default in C:\ProgramData\Seal Report Repository.

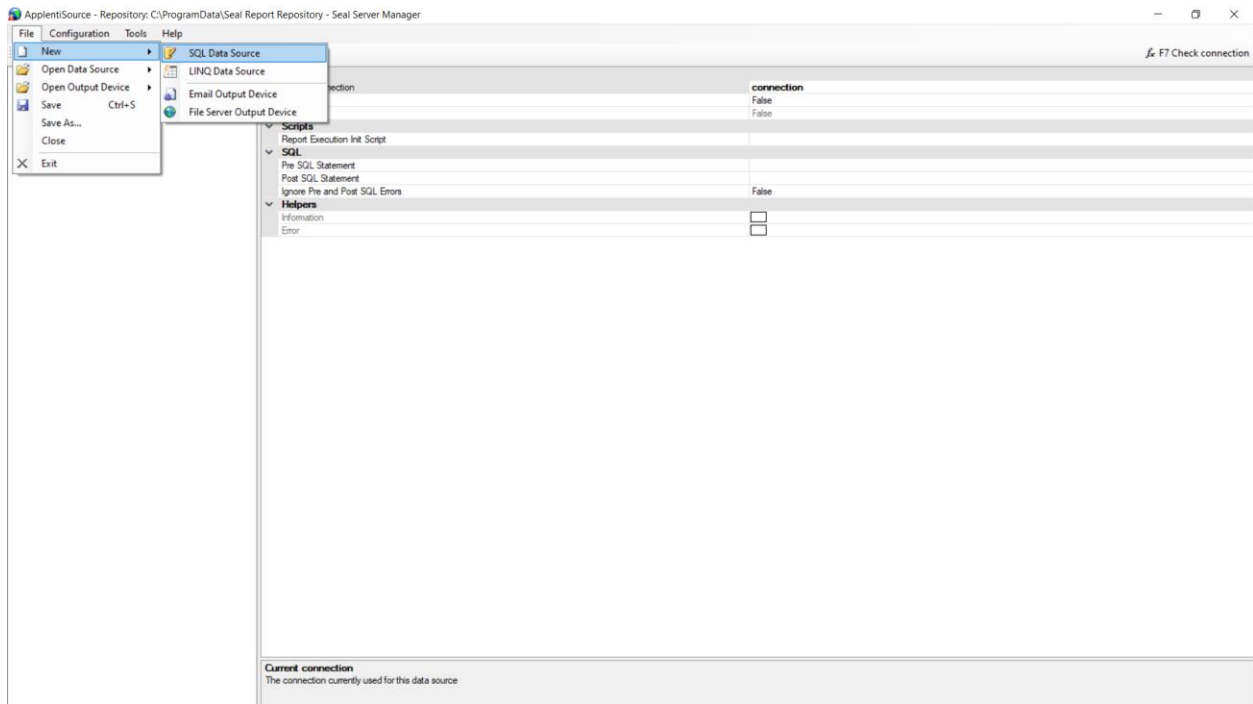
# Steps to follow to make a report

## Step 1 – Adding data source

This can be done either in Report designer or Server manager.

In server manager the following steps are to be followed

File -> New -> SQL Data source



We get the following page after we create our data source.

After a Data source is created, we can add connections to the Database, Add tables, Specify the Joins that we may require and Different types of Enumerated Lists that we can use.

To add a connection, Under the Data source that we created, in Connections -> connection we can set the following values and these values depend on the type of database server i.e. whether it is MS SQL Server or OLE DB etc.

Database type -> 'MS SQL Server'

Connection type -> 'MS SQL Server'

MS SQLServer connection string -> Specify the connection string

The screenshot shows the 'Data Source - Repository: C:\ProgramData\Seal Report Repository - Seal Server Manager' window. The left sidebar shows a tree view with 'Data Source' expanded, containing 'Connections', 'Tables', 'Joins', and 'Enumerated Lists'. The 'Connections' folder is selected, and a 'connection' sub-item is highlighted. The main area displays the configuration for this connection. The 'Definition' section includes fields for Name (connection), Database type (MS SQL Server), Connection type (MS SQL Server), OLE DB Connection string (Provider=SQLOLEDB;data source=localhost;initial catalog=adb;integrated Security=SSPI;), ODBC Connection string, and MS SQLServer Connection string (Server=tcp:applenti-dev.database.windows.net,1433;Database=Enterprise360-Dev;User ID=enterprise360-...;). The 'Date Time format' is set to 'yyyy-MM-dd HH:mm:ss'. The 'Security' section has fields for User name and User password. The 'Helpers' section includes a 'Check connection' checkbox (checked), an 'Information' checkbox, and an 'Error' checkbox. A 'Click to check database connection' button is also present. At the bottom, there is a section titled 'MS SQLServer Connection string' with a description: 'MS SQLServer Connection string used to connect to the database if the connection type is MS SQLServer. The string can contain the keyword '%SEALREPOSITORY%' to specify the repository root folder.'

Definition	
Name	connection
Database type	MS SQL Server
Connection type	MS SQL Server
OLE DB Connection string	Provider=SQLOLEDB;data source=localhost;initial catalog=adb;integrated Security=SSPI;
ODBC Connection string	
MS SQLServer Connection string	Server=tcp:applenti-dev.database.windows.net,1433;Database=Enterprise360-Dev;User ID=enterprise360-...
Date Time format	yyyy-MM-dd HH:mm:ss

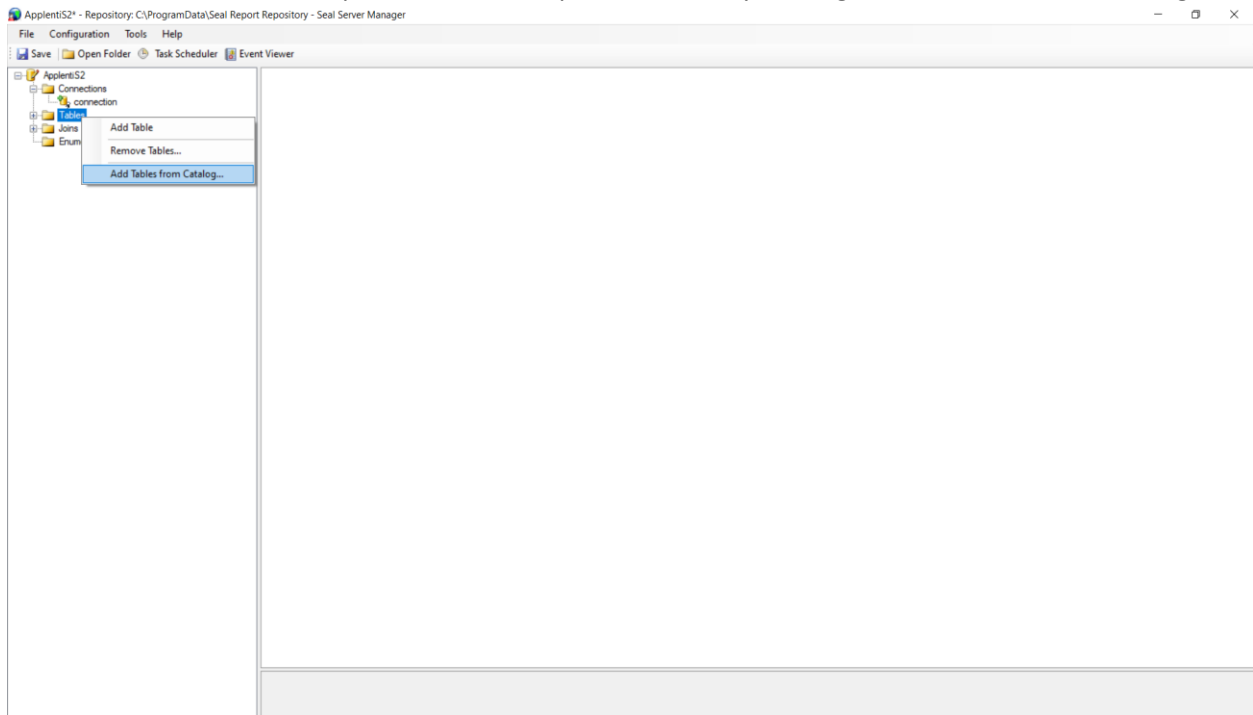
Security	
User name	
User password	

Helpers	
Check connection	<input checked="" type="checkbox"/>
Information	<input type="checkbox"/>
Error	<input type="checkbox"/>

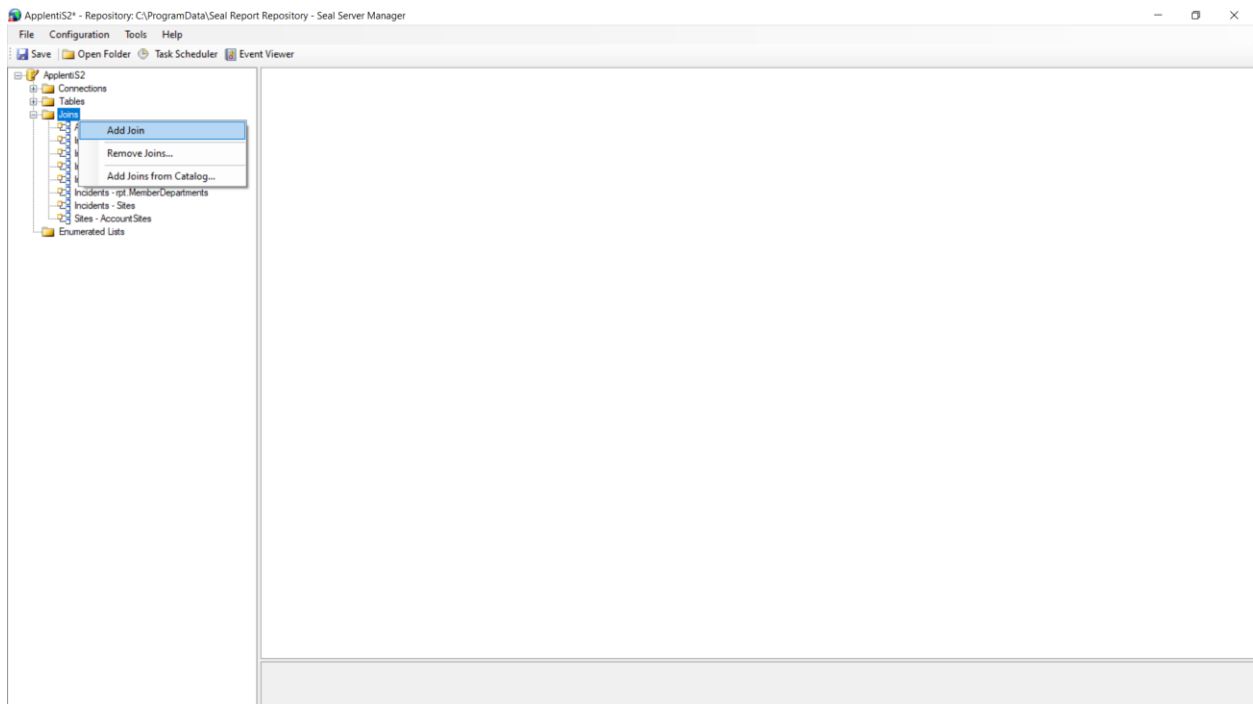
**MS SQLServer Connection string**  
MS SQLServer Connection string used to connect to the database if the connection type is MS SQLServer. The string can contain the keyword '%SEALREPOSITORY%' to specify the repository root folder.

We can check the connection by clicking on "Check connection" present below.

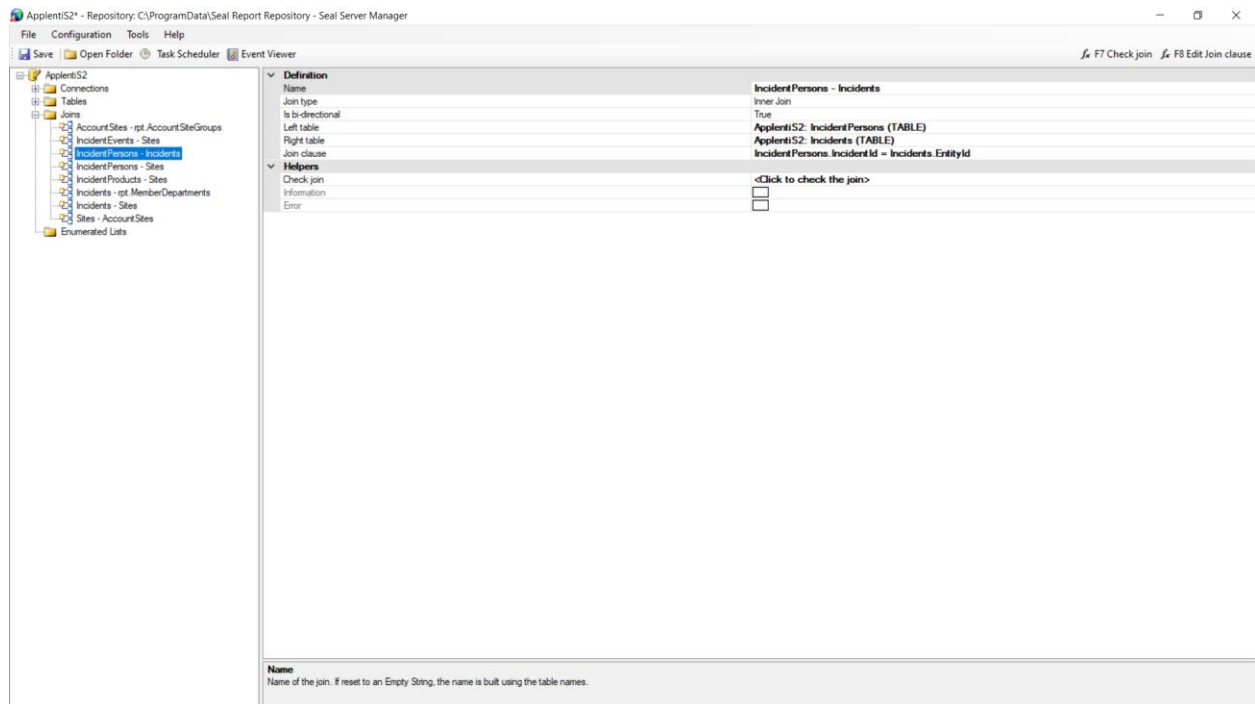
We can add new table or import all the tables present in DB by clicking on 'Add Tables from catalog'



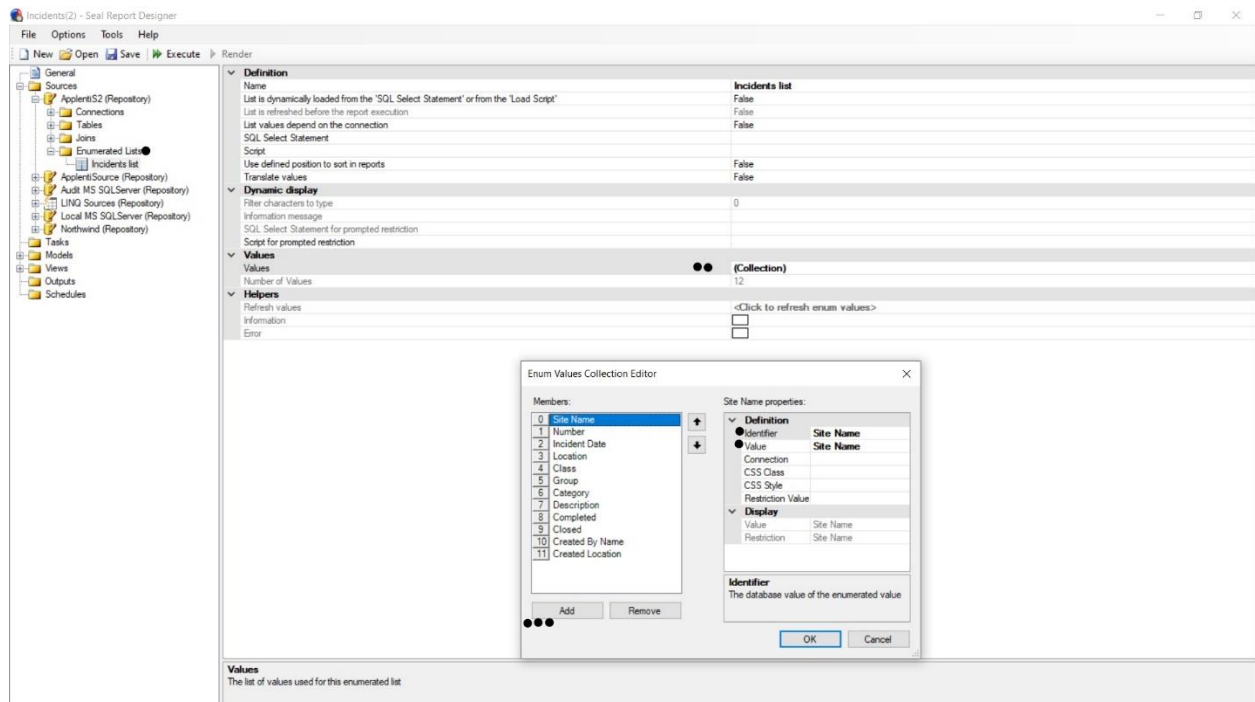
We can also add join or import existing joins by right clicking on Joins



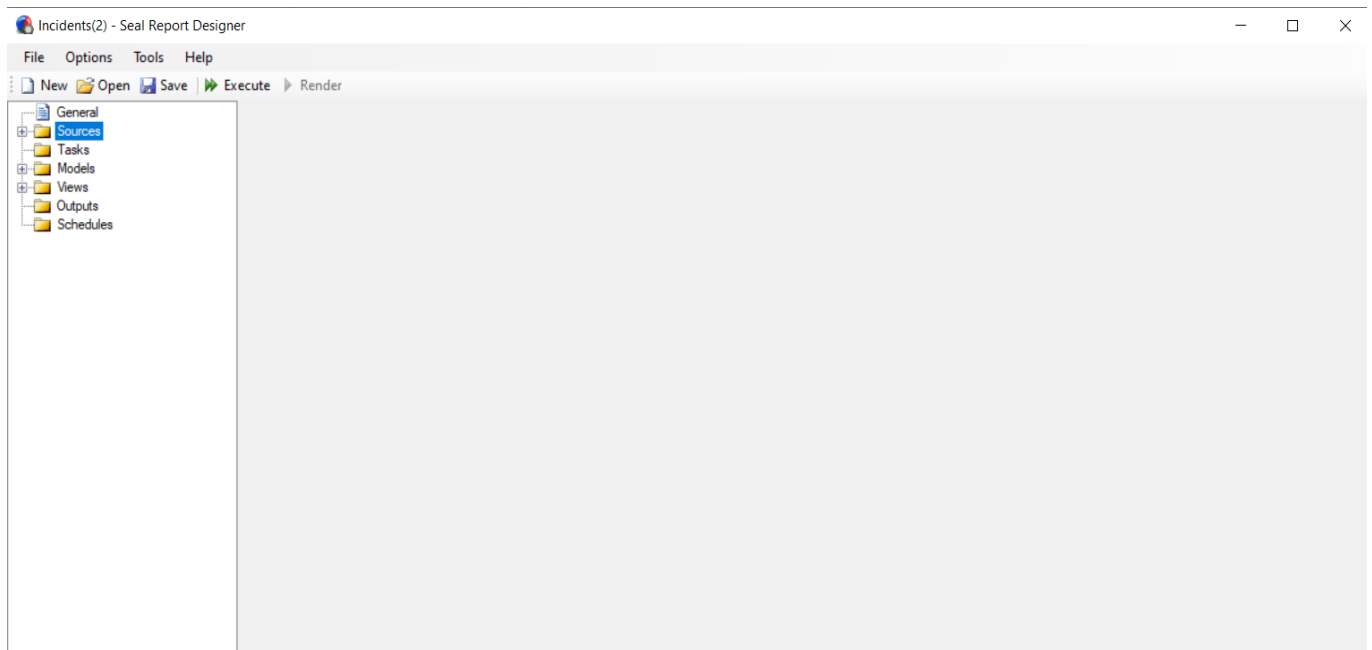
A sample join looks like this. We can specify the type of join in Join type, the tables involved and join clause.



In the Enumerated Lists, we can add a new list in which we can specify the values that we can later use in 'Columns displayed' or 'group by' etc.



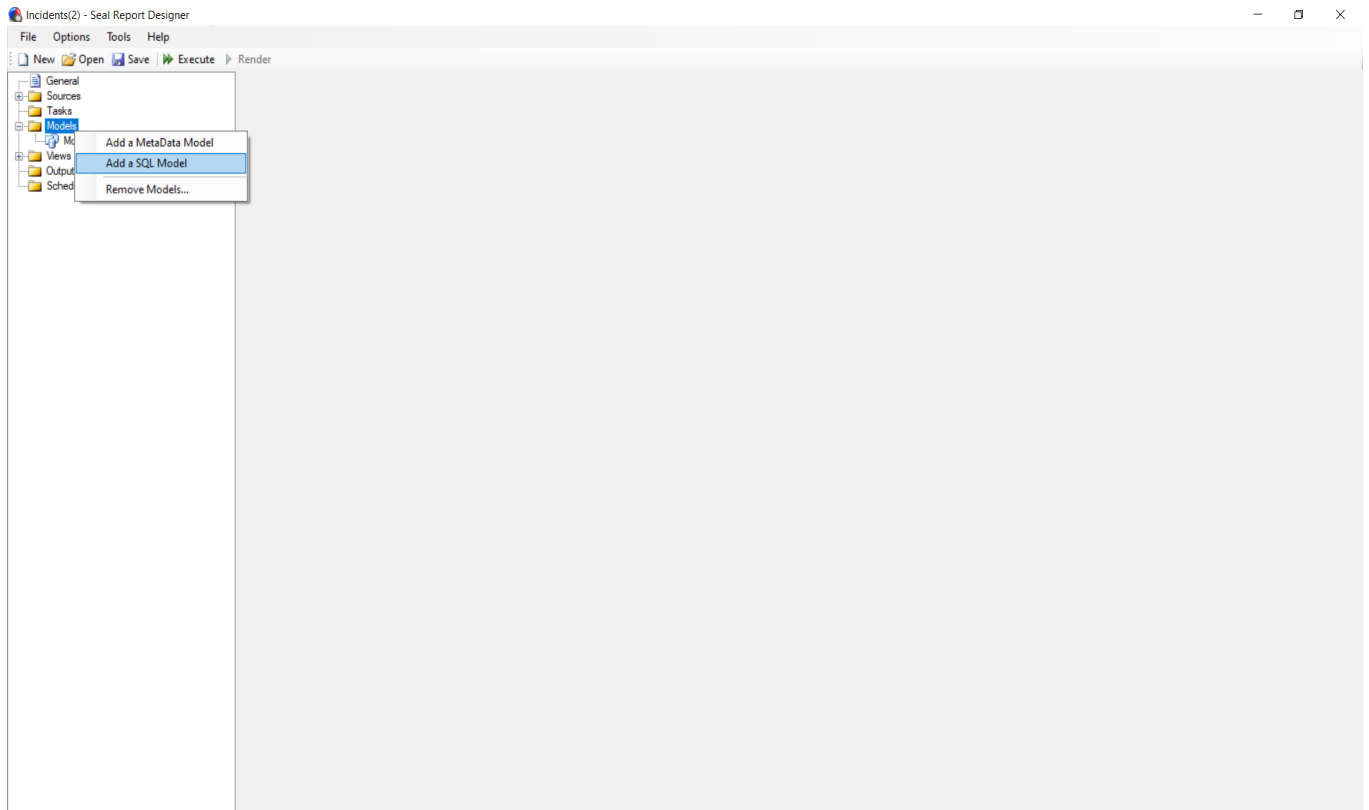
Similar procedure can be followed in Seal Report designer too to create a Source





## Step 2 – Creating a Model

Right click on 'Models' and add the type of model we wish for



After we create our model, the source and connection that the model is gonna use can be specified in Model Definition part.

All the columns that can be used are displayed. We can drag and drop the columns that we require in the Elements section. Depending on how the report should look like we drop the columns in either 'Page elements' or 'Column elements' or 'Row elements' or 'Data elements'

We have added a Pre load script and Post load script in the Model Definition section in the concerned places for “Group by” feature

The following are the scripts we wrote

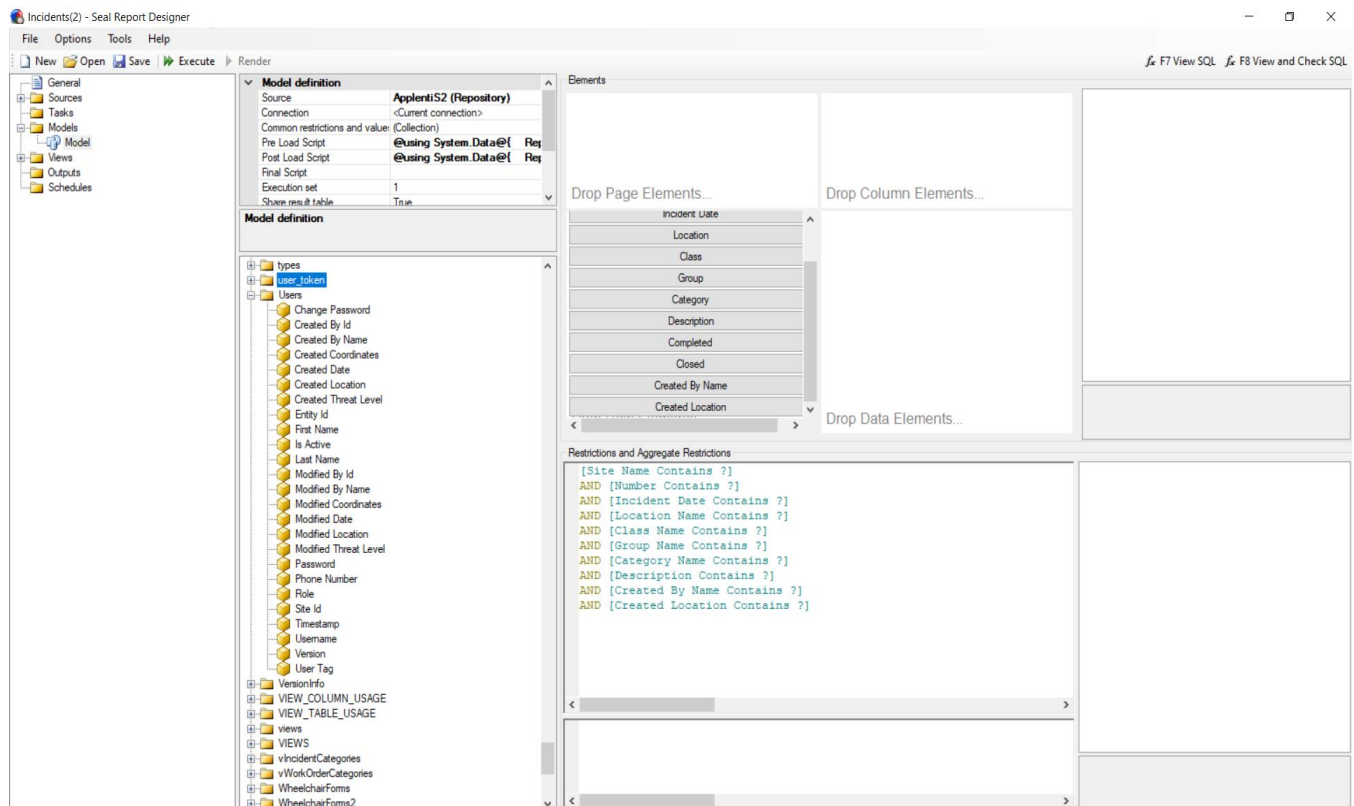
### Pre Script

- Get the restriction from the model
- Check for the previous group by elements if they exist
- Get the new group by values from the restriction
- Find the difference between these 2 arrays
- Now in the last part here we bring these new group by rows to the front of the schema.

### Post Script

- First we follow the above approach to find the new group by values
- Then we find the names of these columns in the actual schema
- Using these names we give a sort order to the result
- Finally we iterate over the result data and group elements until we find new values for the grouped columns since it has already been sorted

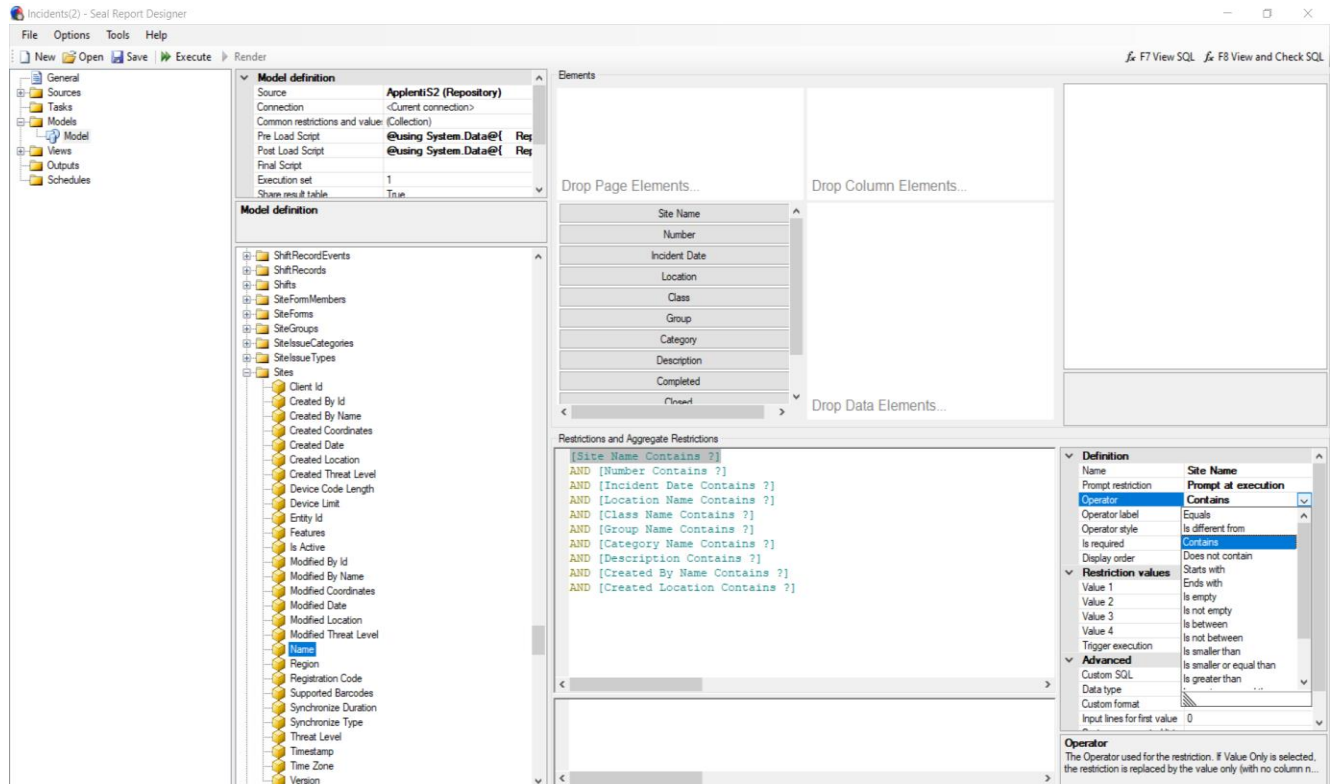
Below Screenshot is for the Incidents report where all the columns are placed appropriately.



The Restrictions on columns can be added by drag and drop of concerned column in the 'Restrictions and Aggregate Restrictions' section.

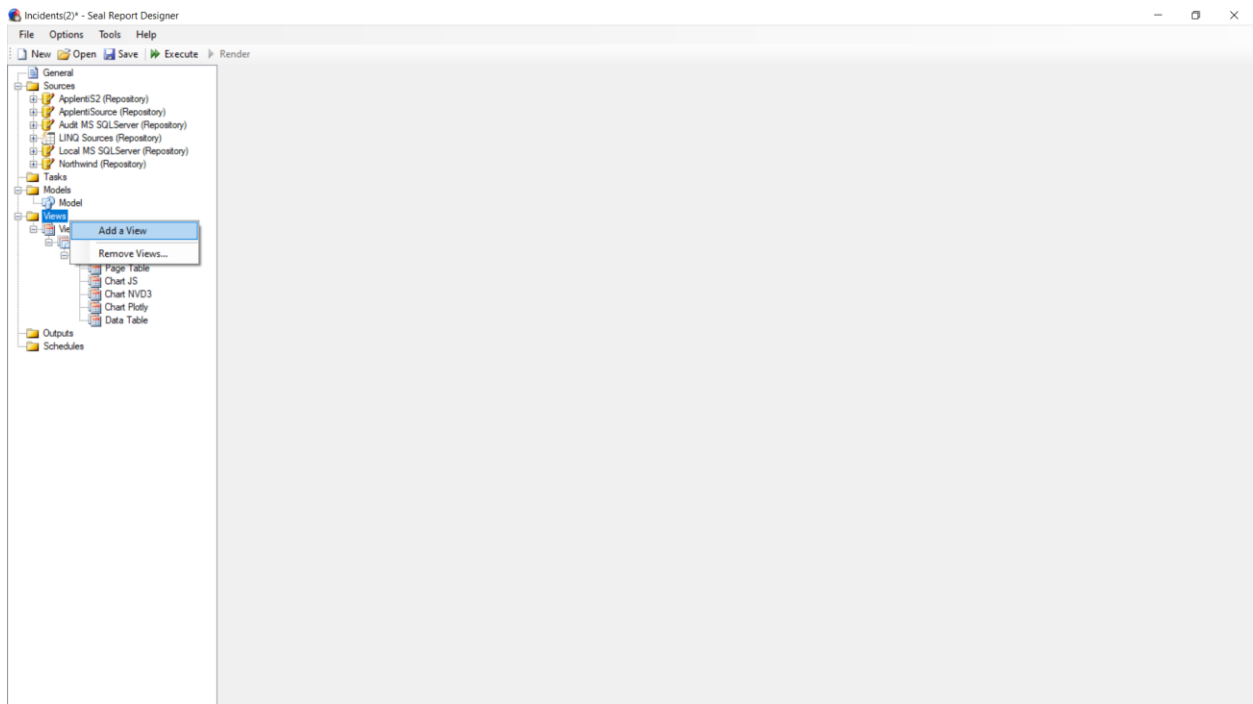
We can set the restriction like 'contains' or 'equals' or 'starts with' etc. as shown in screenshot below.

Also we can make them prompt at execution time and this basically works as a Filter on that column.

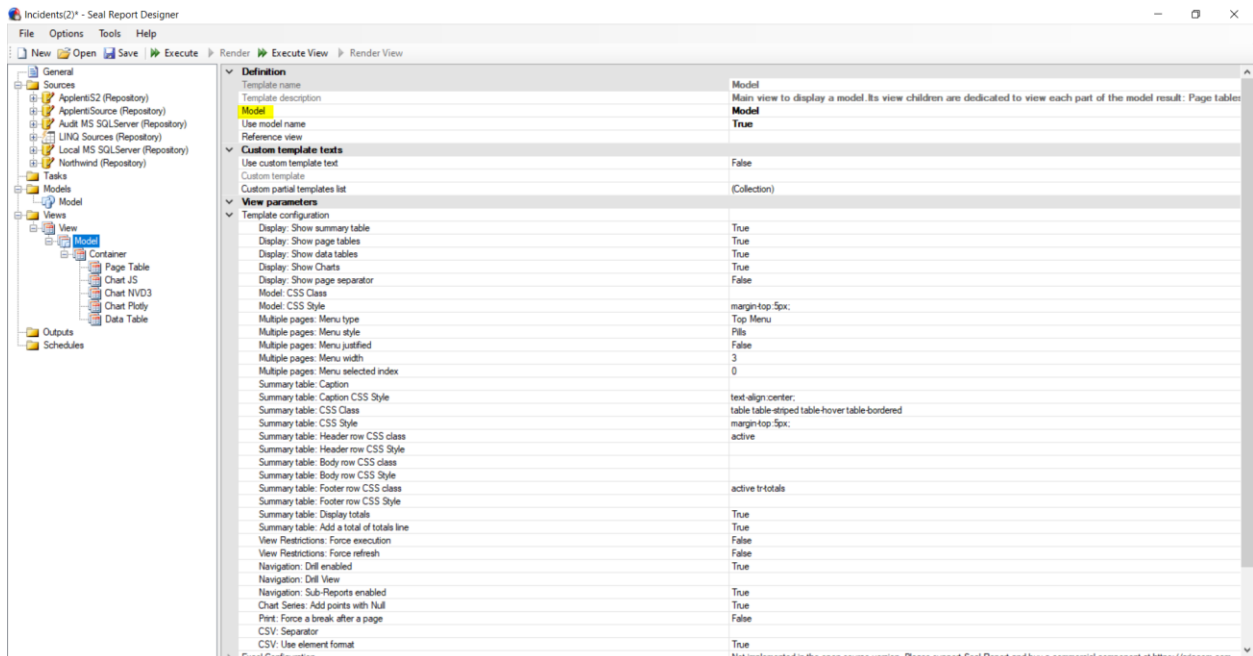


## Step 3 – Create a View

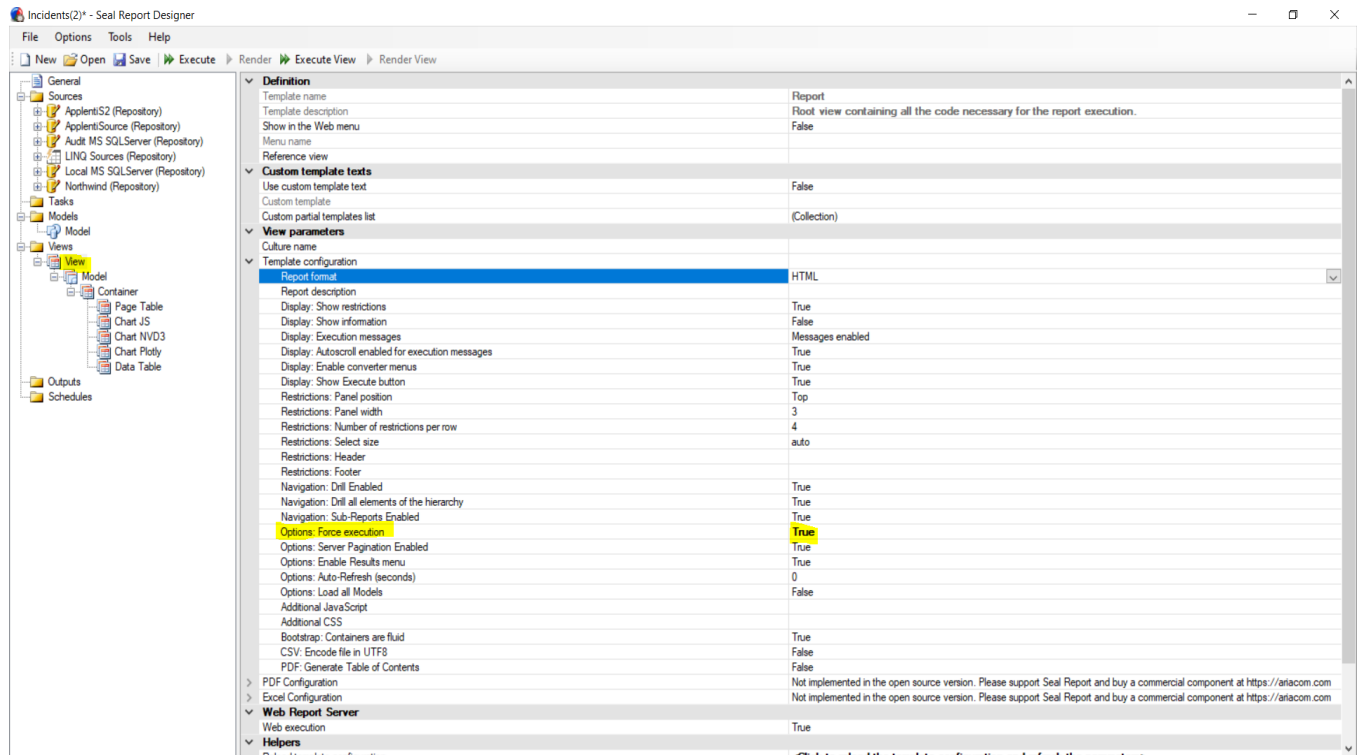
Right click on the view to create a view



Under View -> Model we can select the model we wish to create view for



In View if the option Force execution is set to true then the execution is done when the view loaded without clicking on execute button



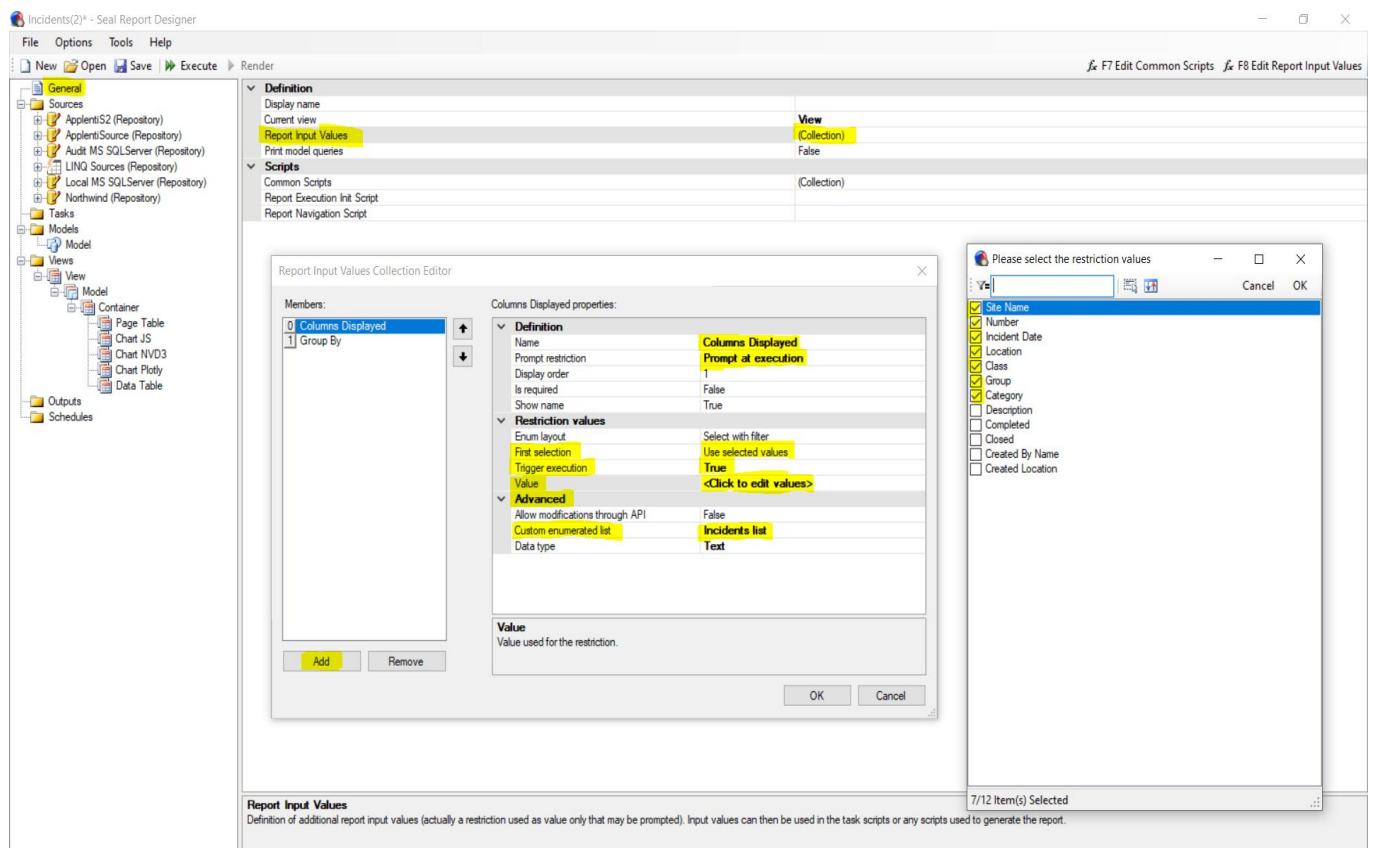
For different views to use some of the common features like 'columns displayed' and 'group by', We can go to General -> Click on 'Collection' under Report Input values -> Add the common elements here.

The values then are set as shown in screenshot.

For the view to load with some default columns, we can select any one of the enumerated lists that we created under 'Advanced' -> Custom enumerated list

After selecting the proper list, under Restriction values -> Value -> <Click to edit values> -> here we can select the columns that are used as default when a view loads

By setting the 'First selection' to 'Use selected values' we will be using only the values that we select and setting 'Trigger execution' to True makes sure the view loads/refreshes whenever a new value is selected.



## Drill down –

In the incident report, if we hover over the incident no, we can see a drill down option to view the ‘information’ related to that incident

Incidents(2) - Seal Report Viewer

Incidents(2)

Columns Displayed: Site Name, Number, Incident Date, Location, Clas

Group By: Nothing selected

Site Name Contains:

Number Contains:

Incident Date Contains:

Location Name Contains:

Class Name Contains:

Group Name Contains:

Category Name Contains:

Description Contains:

Created By Name Contains:

Created Location Contains:

Show 25 rows

Filter:

Site Name	Number	Incident Date	Location	Class	Group	Category
Sample Site 1 - Dev		0001-01-01 00:00:00.0000000	Test8	Class	Group	Category
Sample Site 1 - Dev	IN-0000003	2014-09-15 05:55:08.0000000	Lance - Location 02	Security	Alarms	Alarm Malfunction
Sample Site 1 - Dev	Information	2014-09-15 06:03:36.0000000	Lance - Location 02	Security	Conflict	Assault
Sample Site 1 - Dev	IN-0000041	2014-11-06 13:56:00.0000000	Building 130 - Room 5	Security	Alarms	Alarm
Sample Site 1 - Dev	IN-0000049	2014-11-17 12:07:41.0000000	Lance - Location 02	Security	Conflict	Verbal Abuse
Sample Site 1 - Dev	IN-0000054-001	2014-12-04 14:30:00.0000000	Building 130 - Room 5	Security	Conflict	Injury
Sample Site 1 - Dev	IN-0000077	2014-12-23 07:16:00.0000000	Aztec 130 - Front car park	Security	Alarms	Alarm Malfunction
Sample Site 1 - Dev	IN-0000085-001	2015-01-09 00:01:07.0000000	Aztec 130 - Front car park	Emergency Rescue	Inner region	Floor
Sample Site 1 - Dev	IN-0000100	2015-11-06 21:31:48.7239170	Building 130 - Reception	Emergency Rescue	Outer region	Fire extinguisher
Sample Site 1 - Dev	IN-0000101	2015-11-11 14:01:03.0120060	Building 130 - Reception	Security	Conflict	Assault
Sample Site 1 - Dev	IN-0000102	2015-11-11 17:04:27.0804950	1000KVA Generator Diesel	DK Test	Dk Test - Groop	Dk Test - Category1
Sample Site 1 - Dev	IN-0000103	2015-11-11 20:05:40.1409890	Building 130 - Room 3	DK Test	Dk Test - Groop	Dk Test - Category1
Sample Site 1 - Dev	IN-0000104	2015-11-11 20:20:49.1873450	Building 130 - Reception	Emergency Rescue	Inner region	Floor
Sample Site 1 - Dev	IN-0000105	2015-01-28 03:28:00.0000000	Emergency exit	Security	Alarms	Alarm Vandalized

This is done as shown below

In the model that we created, if we click on the columns that we dropped the elements section then that shows us the table under which the column is actually chosen from. In this case the Incident number column is chosen from the table Incidents, which is highlighted in blue when we click on Number

Incidents(2)\* - Seal Report Designer

File Options Tools Help

New Open Save Execute Render

Model definition

Source: AppletIS2 (Repository)  
Connection: <Current connection>

Post Load Script  
Optional Razor Script to modify the result table of the model just after the...

IncidentProducts

Incidents

Category Name  
Class Name  
Closed By Id  
Closed By Name  
Closed Coordinates  
Closed Date  
Closed Signature  
Closed Signature Url  
Closed Threat Level  
Completed By Id  
Completed By Name  
Completed Coordinates  
Completed Date  
Completed Signature  
Completed Signature Url  
Completed Threat Level  
Created By Id  
Created By Name  
Created Coordinates  
Created Date  
Created Location  
Created Threat Level  
Description  
Entity Id  
Group Name  
Incident Date  
Is Active  
Is Closed  
Is Completed  
Is Revised  
Location Name  
Modified By Id  
Modified By Name  
Modified Coordinates  
Modified Date  
Modified Location  
Modified Threat Level  
**Number**  
Parent Id  
Parent Type

Elements

Drop Page Elements...

Drop Column Elements...

Drop Data Elements...

Restrictions and Aggregate Restrictions

[Site Name Contains ?]  
AND [Number Contains ?]  
AND [Incident Date Contains ?]  
AND [Location Name Contains ?]  
AND [Class Name Contains ?]  
AND [Group Name Contains ?]  
AND [Category Name Contains ?]  
AND [Description Contains ?]  
AND [Created By Name Contains ?]  
AND [Created Location Contains ?]

Definition

Name	Number
Sort order	Automatic Ascendant

Chart

Options	
Data type	Default
Show sub-totals	False
Custom format	

Advanced

Custom SQL	
Cell script	
Cell navigation script	
Custom enumerated list	
Force aggregate	Default
Contains HTML	False

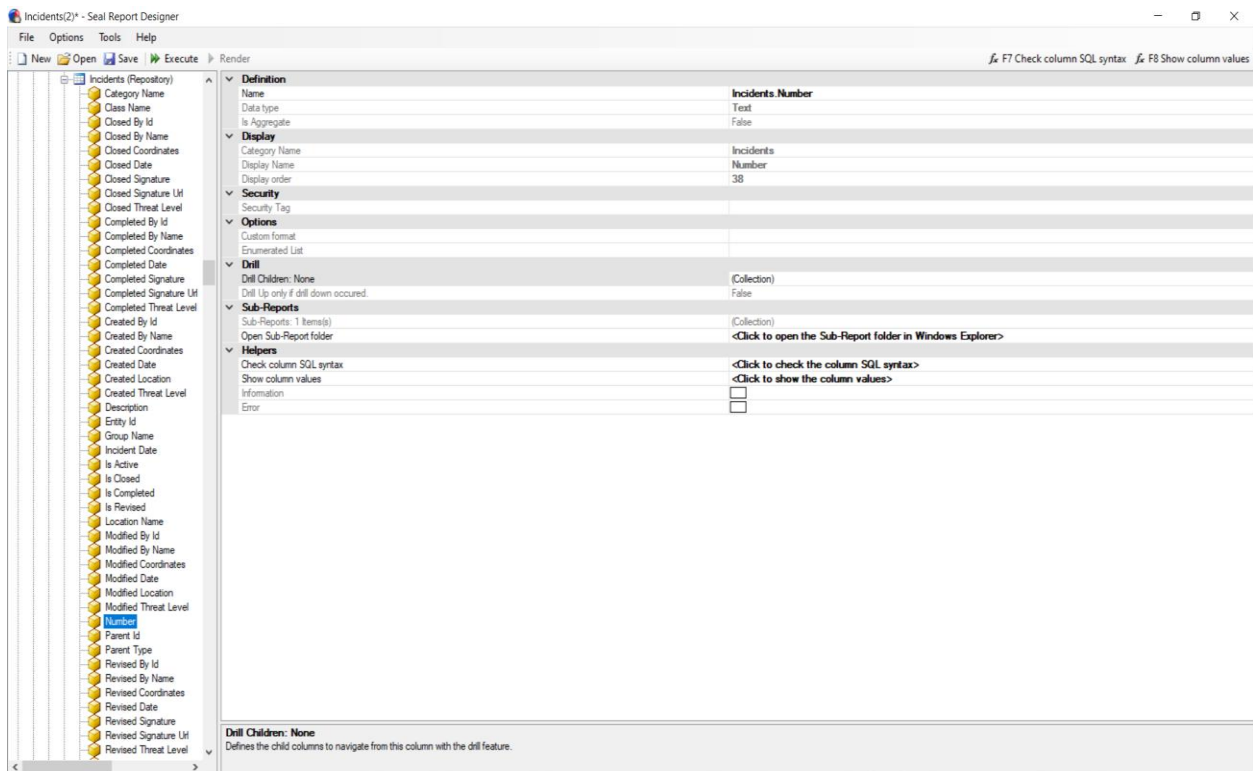
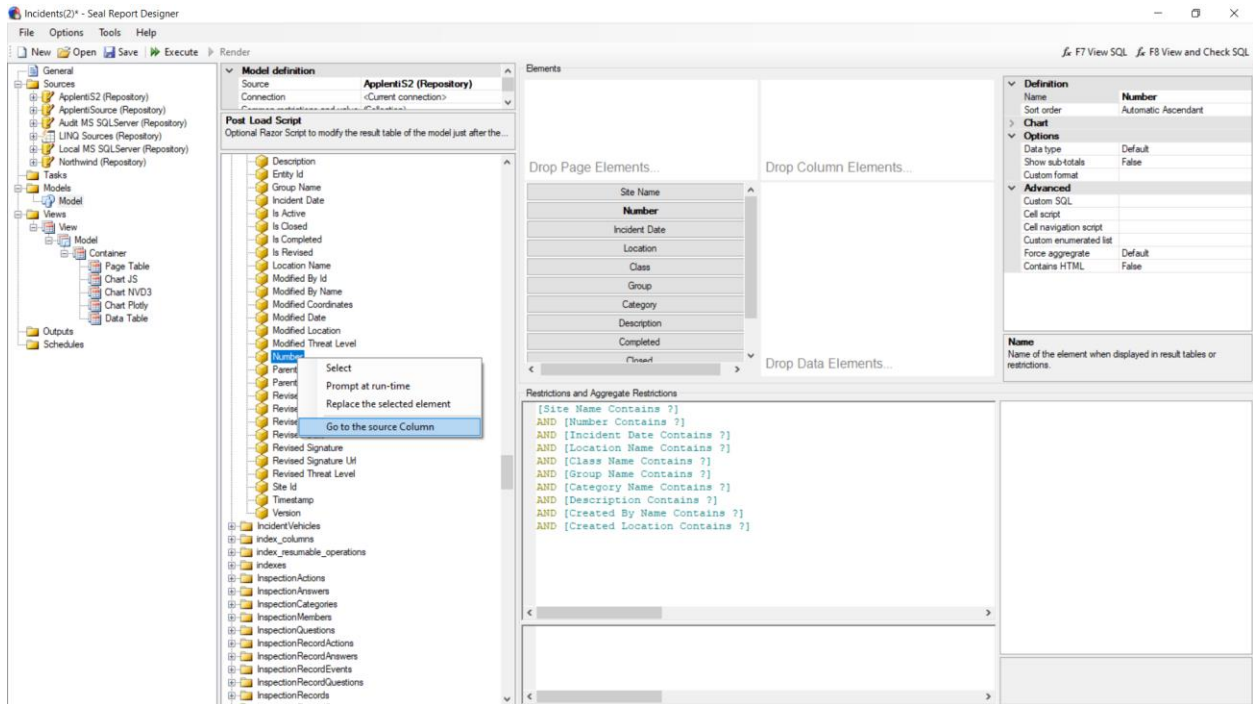
Name

Name of the element when displayed in result tables or restrictions.



Right click on the column and click on 'Go to Source Column'

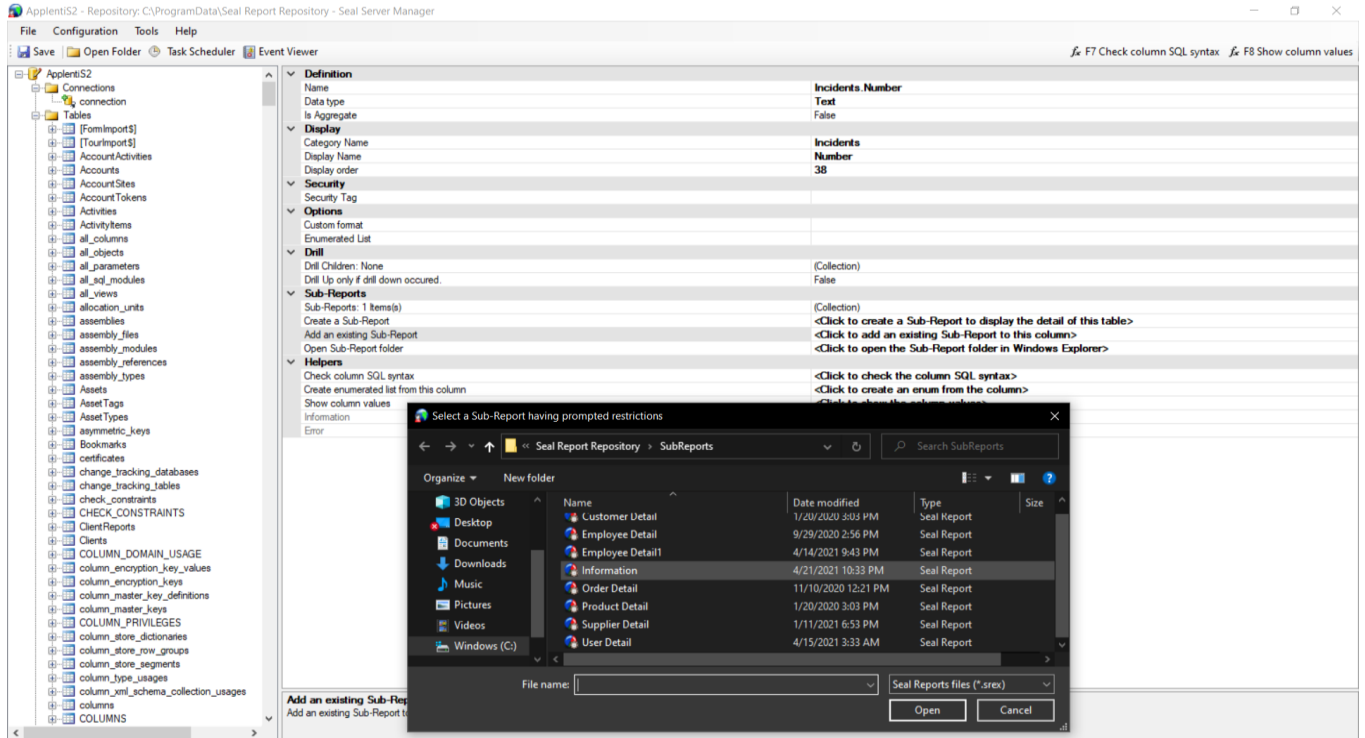
This takes us to the data source under which the table of the selected column exists



So to generate a sub report when we click on the 'Information' that we see in our view, we have to add a 'Sub report' under 'Sub reports' as seen in above screenshot. This is done in Server manager

In Sever manager -> open the data source that we created -> go to tables -> Incidents -> Number

Here we can just attach the report we want to be shown when we drill down in to the Incident number of our previous report. This is done by creating another report and adding it under 'Sub-reports' -> Add an existing sub report -> and in this case it is the 'Information' report.



The screenshot displays the Seel Report Designer application. The top menu bar includes File, Options, Tools, and Help. Below the menu is a toolbar with icons for New, Open, Save, Execute, and Render. The main workspace is divided into several panels:

- Left Panel (Tree View):** Shows a hierarchical structure of the report model. The 'Models' folder is expanded, showing 'Description', 'Model', 'Model2', 'Persons', and 'Signature'. The 'Views' folder is also expanded, showing 'View', 'Information', 'Description', 'Persons', 'Container', 'Page Table', 'Chart JS', 'Chart NVD3', 'Chart Plotly', and 'Data Table'.
- Model definition Panel:** Displays the 'Model definition' for 'ApprentiS2 (Repository)'. It includes fields for Source, Connection, Common restrictions and values, Pre Load Script, Post Load Script, Final Script, Execution set, and Show result table. Below this, the 'Source' section lists the source used to build the model, including a list of tables and views.
- Elements Panel:** Contains four sections: 'Drop Page Elements...', 'Drop Column Elements...', 'Drop Row Elements...', and 'Drop Data Elements...'. Each section has a list of elements to be dropped into the report.
- Restrictions and Aggregate Restrictions Panel:** Displays the 'Restrictions and Aggregate Restrictions' for the selected model. It includes a text area for restrictions and a list of aggregate restrictions.

The following report is generated when we click on Incident number -> Information of our report

[illegible]

Suspect P Three 0

Three

#### Injuries

Full Name	Injury Type	Body Part	Notes	Created Date
-----------	-------------	-----------	-------	--------------

#### Signature

Completed By	Completed Date	Revised By	Revised Date	Closed By	Closed Date
Vicente Guzman	11/17/2020 11:57 AM		01/01/0001 12:00 AM		01/01/0001 12:00 AM

#### Pictures

Image Uri	Notes
-----------	-------



#### Events

#### Products

Description	Stock Code	Quantity	Value	Notes	Created By Name
-------------	------------	----------	-------	-------	-----------------

#### Vehicles

Plate	Make	Model	Color	Notes	Vehicle Type	Year	Decal	Created By Name
-------	------	-------	-------	-------	--------------	------	-------	-----------------

#### Versions

Number
IN-0000268-005
IN-0000268-004
IN-0000268-003
IN-0000268-002
IN-0000268-001

## Deployment as Azure Service

1. Create the Azure App Service (.NET core for Linux app, .Net for Windows app)
2. From the Report Designer tool edit the report System\500 Publication - Web Report Server  
Modify the task script 'Publish Web Report Server' to configure the Destination Directory to '/site/wwwroot/':

```
webServerDestination = @"/site/wwwroot/";
```

Modify the script to configure the FTP Client (Host Name, User Name, Password) using the information got from Azure App Service.

```
client = new FtpClient("waws-prod-dm1-163.ftp.azurewebsites.windows.net",  
@ "sealcore\sealcore", "password got from the Azure portal");
```

3. Execute the report

Set the 'Web Report Server Distribution' restriction: Select .NET Core for Linux App and .Net for Windows App and Re-execute the report. Files should be copied from the installation directory '\NETCore' or '\Web.Net' Sub-Folder to the remote directory '/site/wwwroot/'

4. Edit the report System\510 Publication - Repository and Modify webServerDestination and client details as previous step and execute report

Set the restriction 'Repository Source Folder' to 'Repository: Copy All Folders'.

Re-execute the report.

Repository Files should be copied to the remote directory '/site/wwwroot/Repository/'.

## 5. Configure the App Service

In Azure Portal, in configuration of created app service, setup startup command

- .NET core, Linux : 'dotnet /home/site/wwwroot/SealWebServer.dll'
- .Net, Windows : Depending on your distribution, set the Platform to '32bit' or '64bit'

Go back to Overview and click [Browse] to test the site

## Deploying New Reports to existing AppService:

1. All main Reports file should be copied to /Repository/Reports/
2. All SubReports file should be copied to /Repository/SubReports
3. Restart the app service

## Front End

1. Use the html to view reports in iframe
2. Currently "Users" and "Incidents" reports are already deployed to <https://applenti-sealreports-dev.azurewebsites.net/> - could be changed to your service
3. Open the frontend/hostingstart.html page in browser
4. Each button on click executes and fetches reports as specified
5. Multiple options are provided for filtering and aggregation
6. Export csv option should be available to download report

## Integration note in Azure

1. Authentication is not used and hence it is important to set few configuration parameters after deployment to be able to fetch reports in iframe
2. Access deployed service folder /site/wwwroot, and update the file web.config to following in <system.web> configuration

## Things to be done

Below are some points to be taken care of in the reports/ sub-reports:

- 1) Format Time based on the Time zone of the sites.
- 2) For images, add custom script in the cell script for the column which has the image URL.
- 3) To display all the versions of the Incident, add script in pre-load script so that the restriction will check for the columns which contain the first 11 characters of the Incidents number.

Below are some points which are yet to be developed.

- 1) Export to PDF feature – Export to CSV is a feature which is provided by Seal for free. Export to PDF is to be developed.
- 2) Passing Parameters – The feature to pass parameters like site id, account id and client id as input to the SQL queries is to be developed.
- 3) Bookmark Creation – The feature to save the report as bookmark is also not developed.
- 4) Authentication is not handled, while integrating with other application, it is important to handle session parameters in report tool for security