

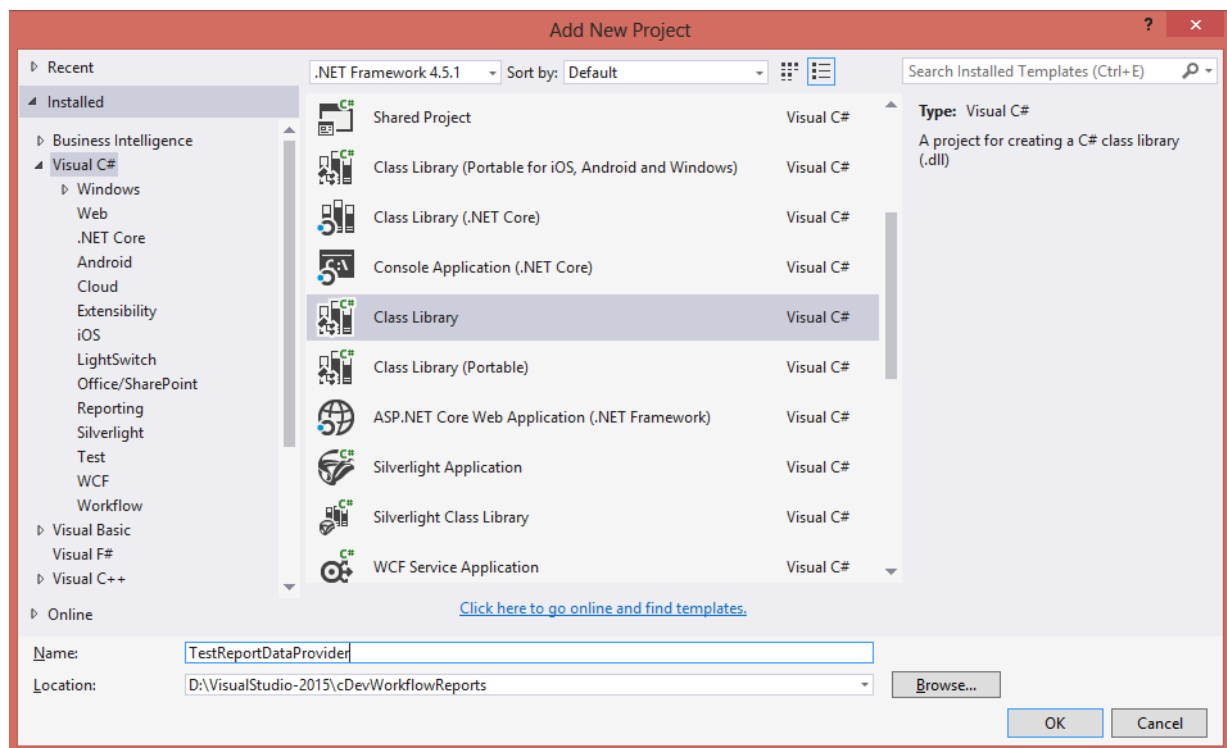
## **Building, Configuring and Displaying SSRS Report with Data-Provider**

This sample will illustrate how to write a new report data provider, how to configure it with SSRS report file using the Configuration Manager (Reports), add the report definition and running the report.

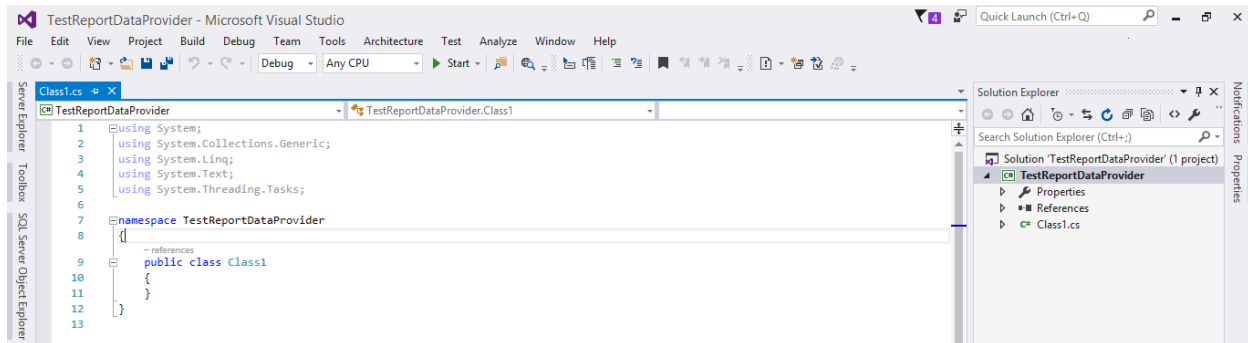
### Build a Data Provider in Visual Studio

Start Microsoft Visual Studio 2015 and select “New Project”. Select “C#” as the language and select “Class Library” as the project type. This example is going to use C# as the development language. This same example can be also written using Visual Basic .Net.

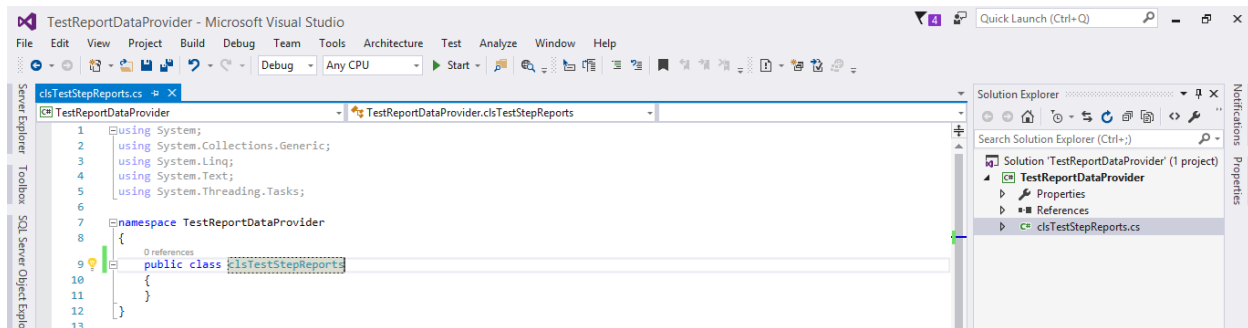
For the name of the project, provide “TestReportDataProvider”.



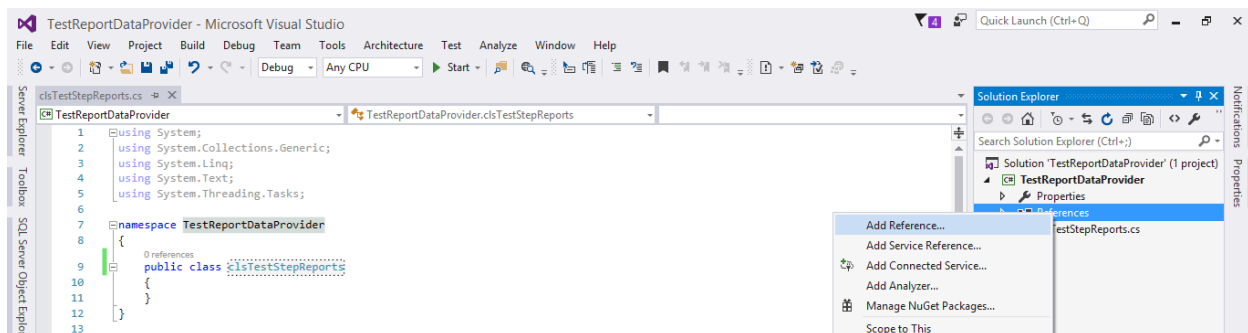
After clicking , OK the solution opens up.



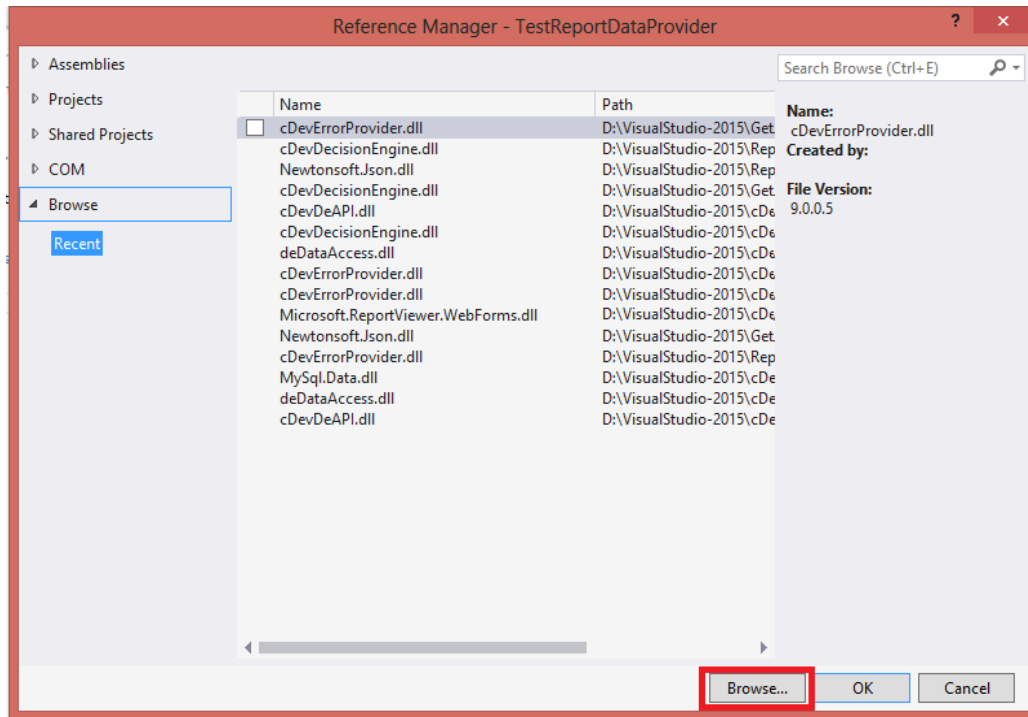
Once the project is created, select “Class1.cs” and rename the class to “clsTestStepReports.cs”



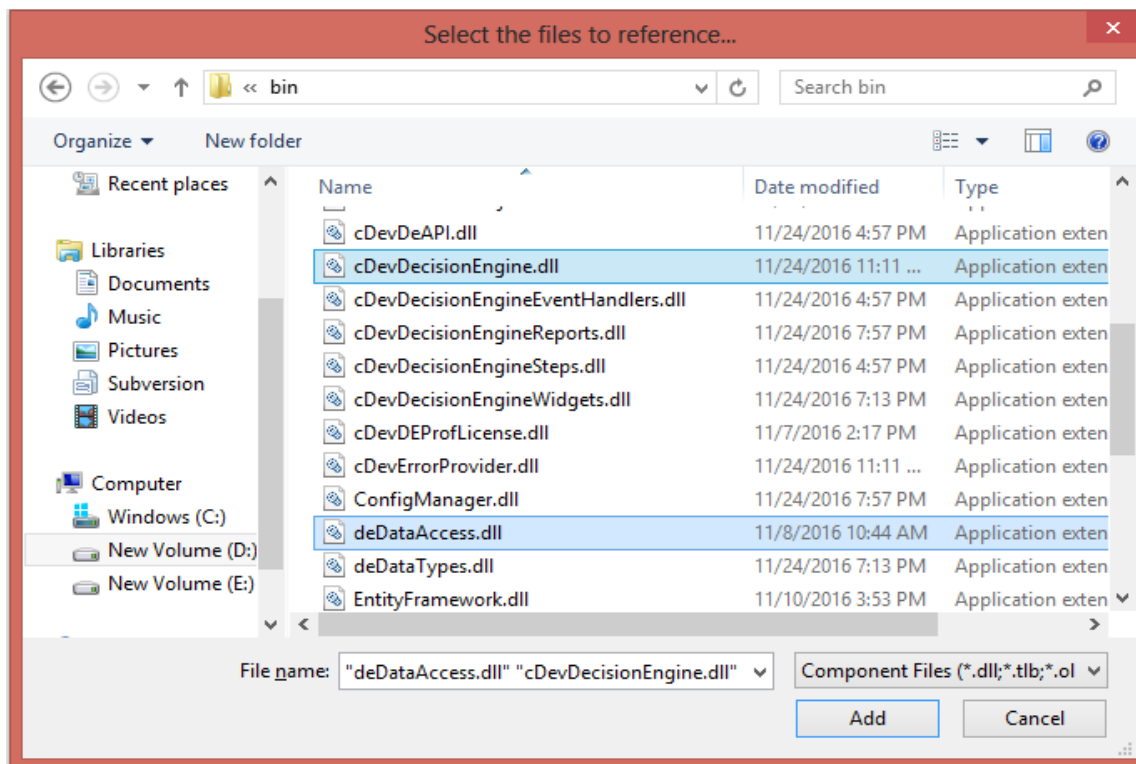
Right click References under TestReportDataProvider Project. Click Add Reference...



Reference manager window will open. Click browse button.

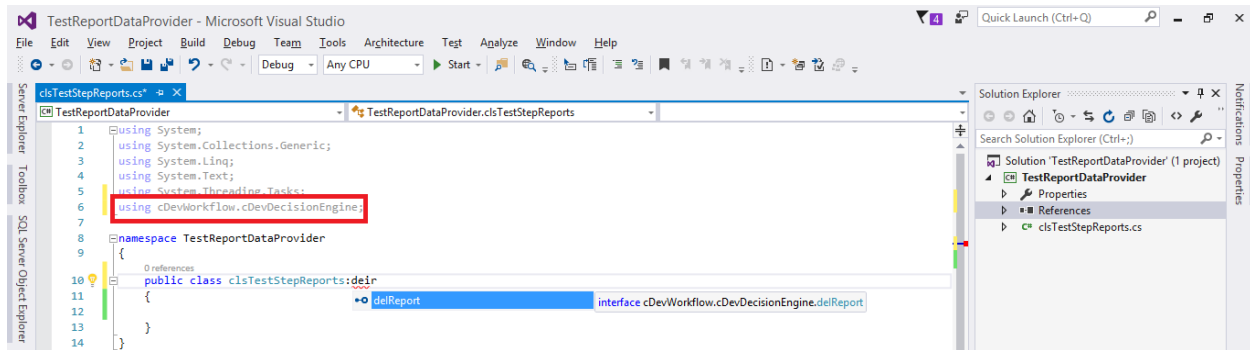


Add a reference to cDevWorkflow, navigate to the following directory  
 “C:\inetpub\wwwroot\cDevWorkflow\bin”. Select the files “cDevDecisionEngine.dll , deDataAccess.dll”.



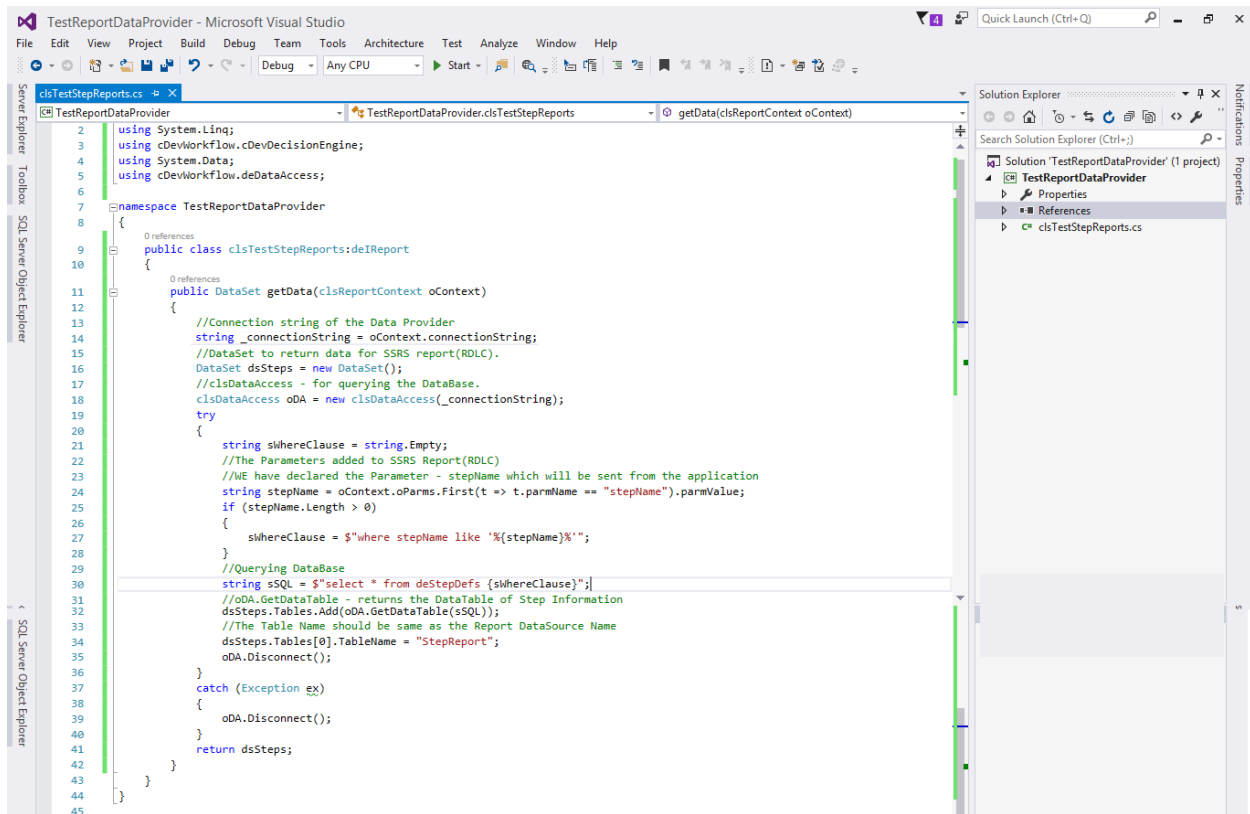
Add a using statement : cDevWorkflow.cDevDecisoinEngine

Let's start building the class, first implement the interface called: "deIReport".

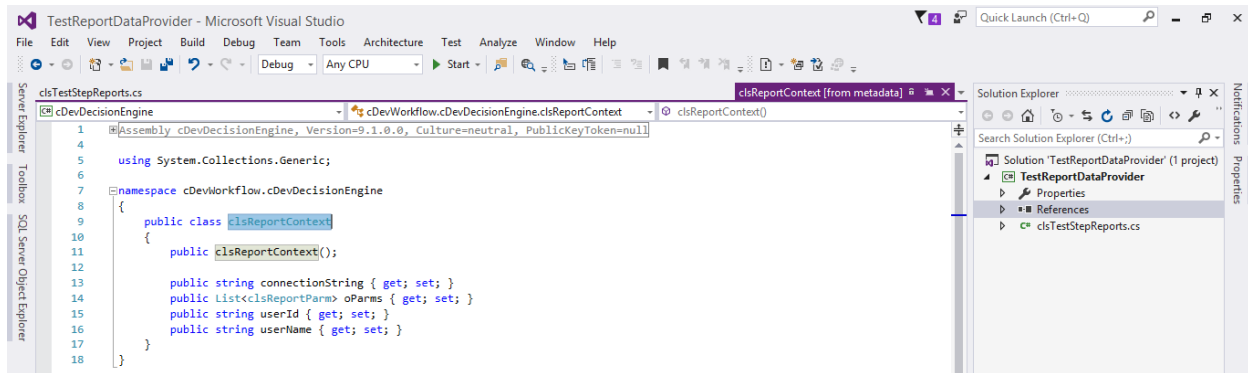


In order to implement the interface, we have to implement the "getData" method. The getData method is going to return DataSet object. DataSet has data to bind to SSRS report.

The clsReportContext is the input for the method getData.



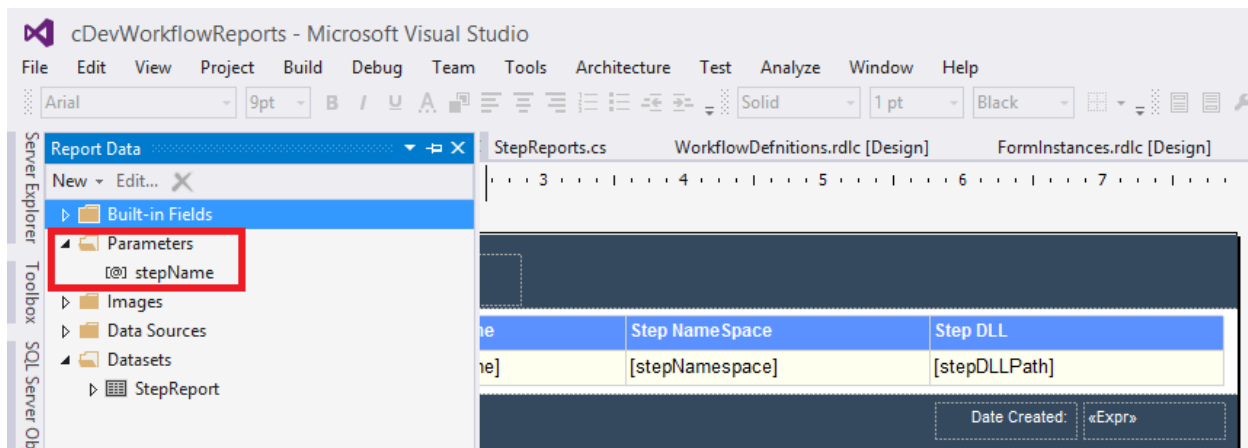
The clsReportContext contains the parameters list,connection string,userid and username.



Using clsDataAccess , We can write our own query to fetch data from database.

The important sections need to be focused on are the Input parameters and the Tablename given for the DataSet each datatable.

The Input parameters will be read from the SSRS(RDLC) Report parameters.



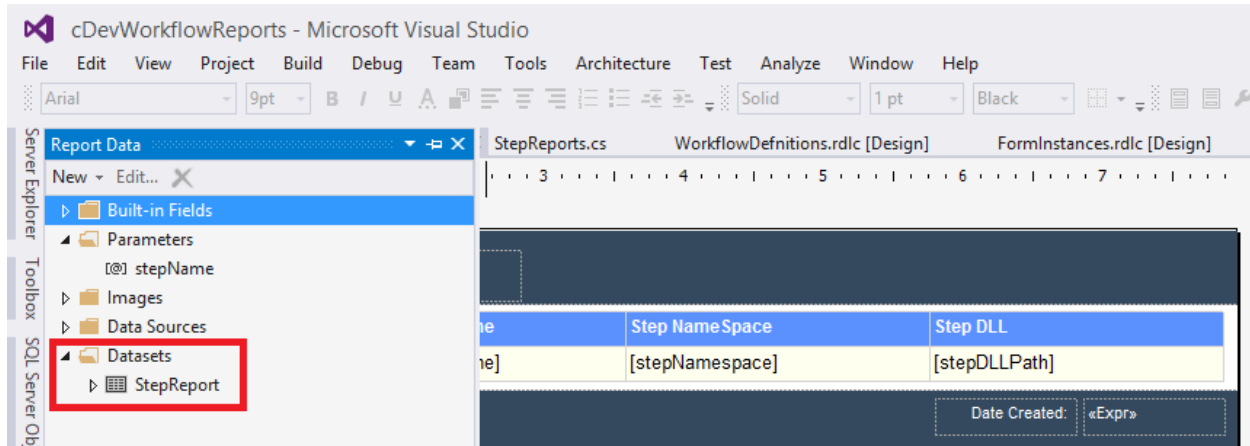
Here, we have to capture the parameter value and bind it to query (if query expects parameter) using Context.oParams .

```

//The Parameters added to SSRS Report(RDLC)
//WE have declared the Parameter - stepName which will be sent from the application
string stepName = oContext.oParams.First(t => t.parmName == "stepName").parmValue;
    
```

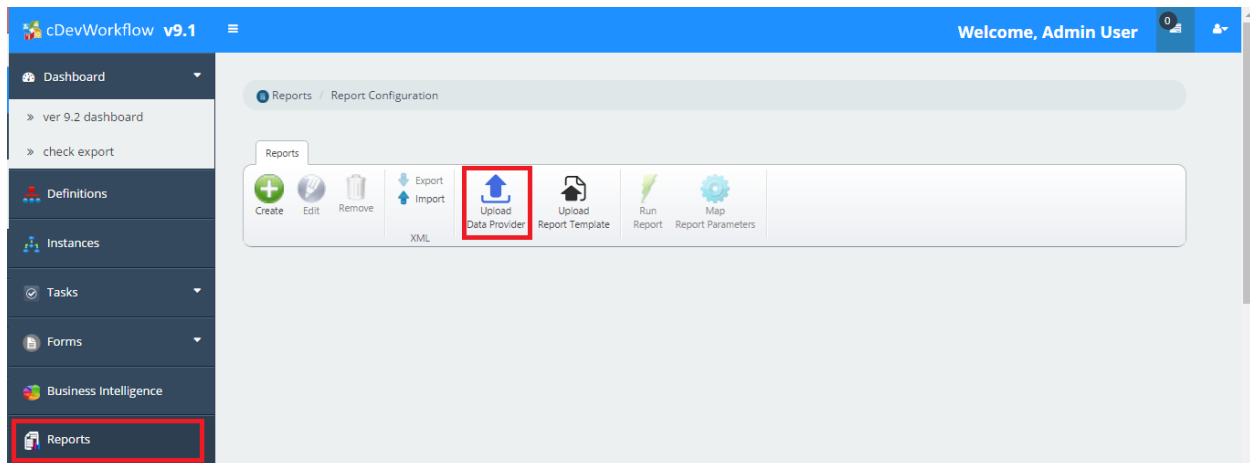
The TableName given for each datatable(s) of dataset should be same as the Name assigned to the datasets during the design of SSRS(RDLC) report.

```
//The Table Name should be same as the Report DataSet(s) Name
dsSteps.Tables[0].TableName = "StepReport";
```

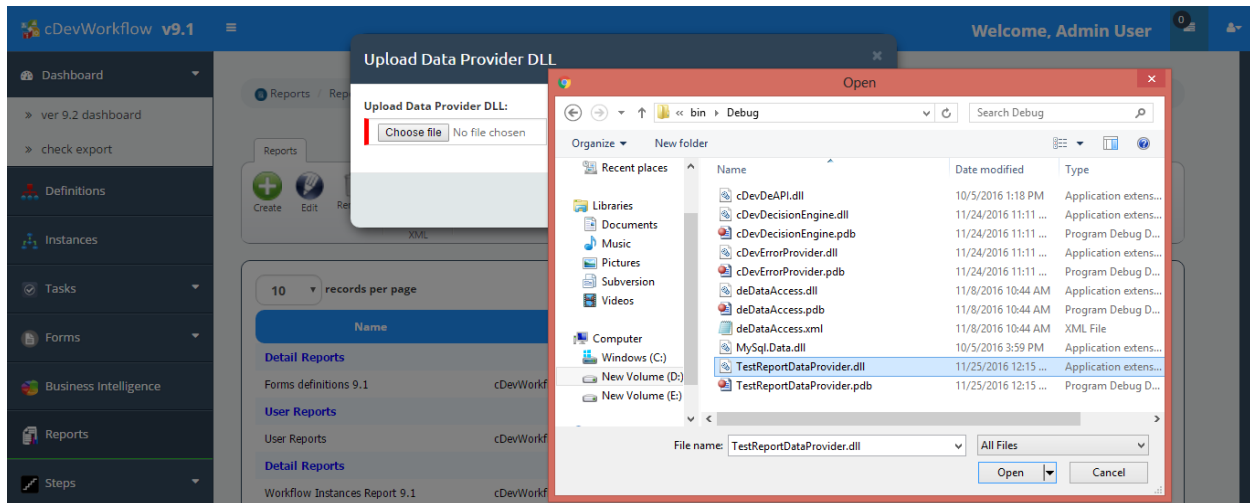


## Upload Data Provider

After, Building the Data-Provider. Upload the (Data-Provider (.dll)) using, the Upload Data-Provider provided in the Reports section of cDevWorkFlow.

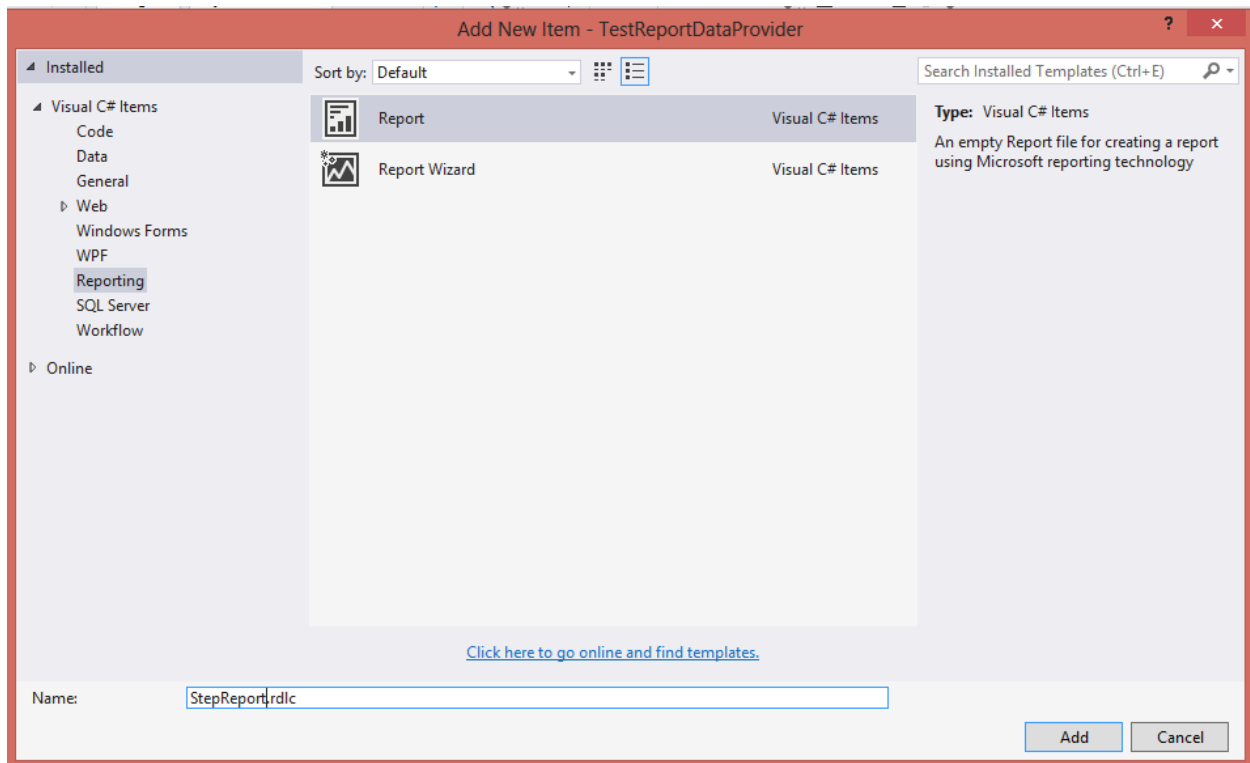


After Clicking, “Upload Data Provider” select “TestReportDataProvider.dll”

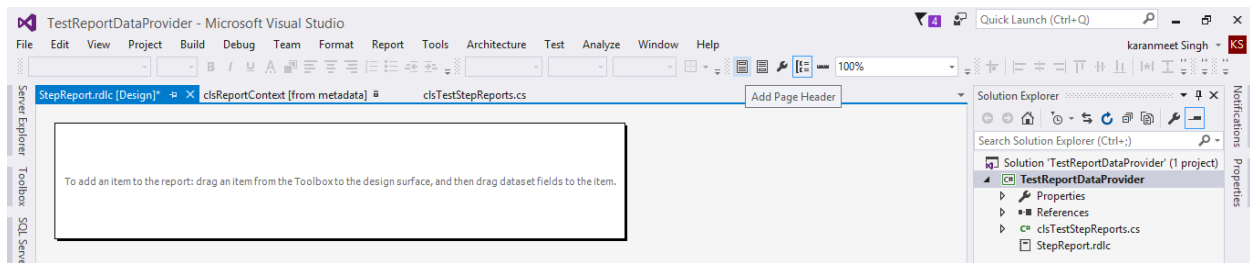


## Build Report Template in Visual Studio

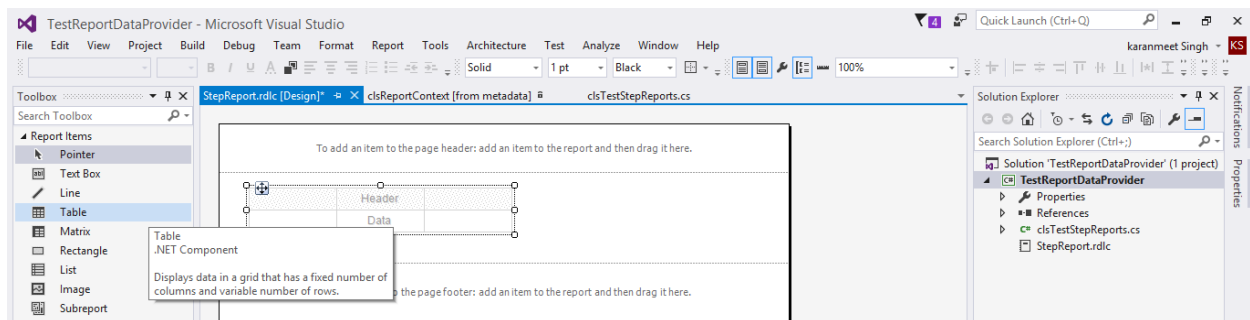
Open Visual Studio-> In the same Solution "TestDataProvider" -> Add Item -> Reporting -> Report->StepReport.rdlc



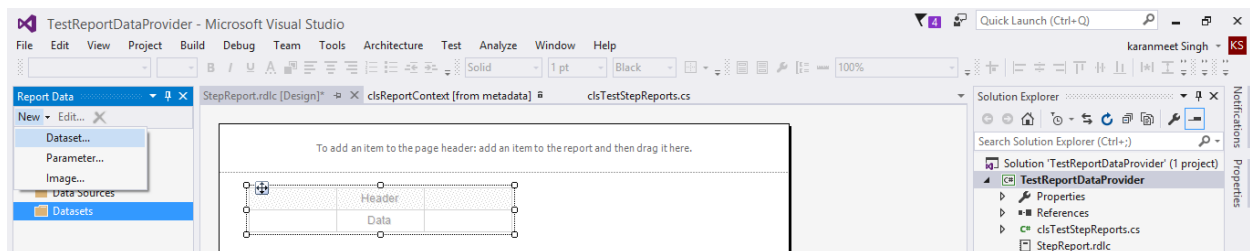
## Add Header and Footer (If required)



## Add Table from ToolBox



## Select Add DataSet (For Creating Report Fields in SSRS(RDLC) file)



## Right Click DataSets-> Add DataSet



Dataset Properties

General

Choose the Dataset

Name:  
StepReport

Data source:  
StepReport

New...

Available datasets:  
deStepDefs

Fields:

Field Name	Type Name
stepID	System.Guid
stepName	System.String
stepDescription	System.String
stepIndex	System.Int16
stepCategory	System.Guid
stepNamespace	System.String
stepDLLPath	System.String
defCount	System.Int16
stepDisplayName	System.String

Help OK Cancel

## After DataSet Selection

New Edit...

Built-in Fields

Parameters

Images

Data Sources

Datasets

StepReport

stepID

stepName

stepDescription

stepIndex

stepCategory

stepNamespace

stepDLLPath

defCount

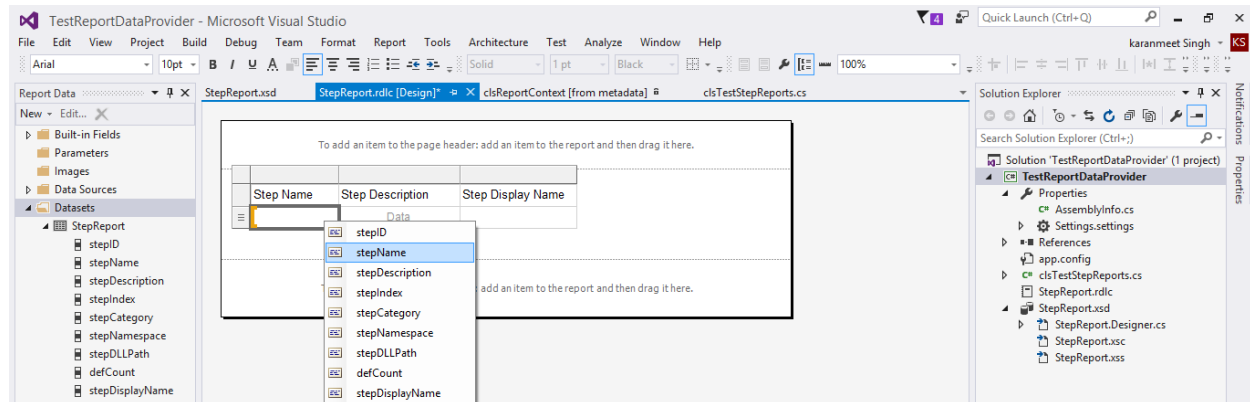
stepDisplayName

To add an item to the page header: add an item to the report and then drag it here.

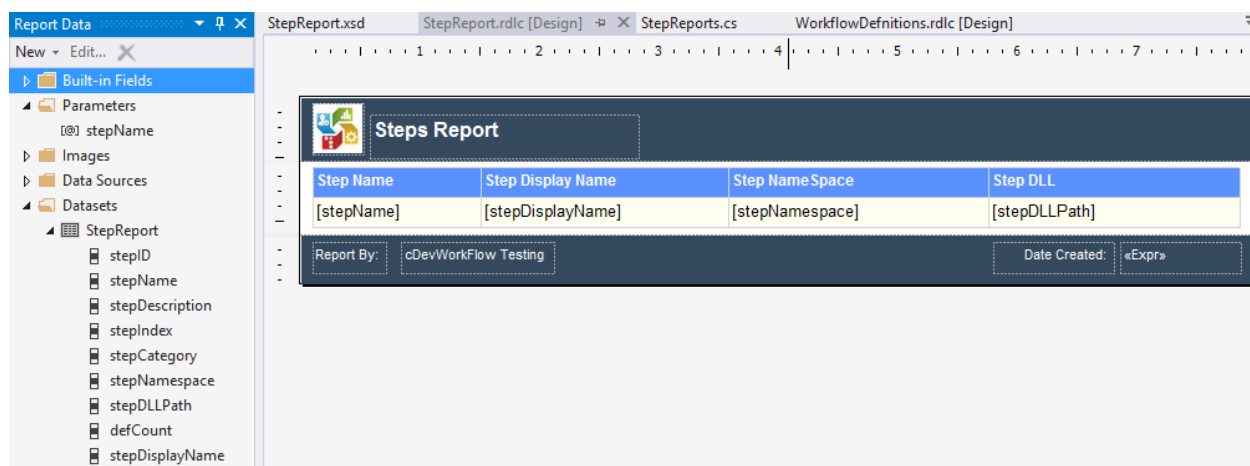
Header
Data

To add an item to the page footer: add an item to the report and then drag it here.

## Select the Required DataTable column to the Table added



The Report RDLC file can be designed. After, Design the Report looks.



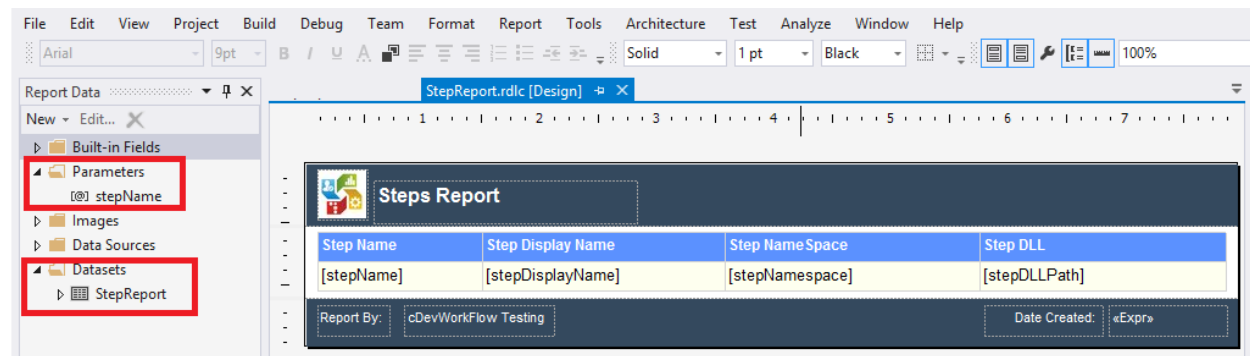
## Upload Report Template

Build required SSRS (RDLC) Report Template with required datasets and Parameter(If required).

For ex: Please consider the below SSRS (RDLC) Report Template for Step Information. With

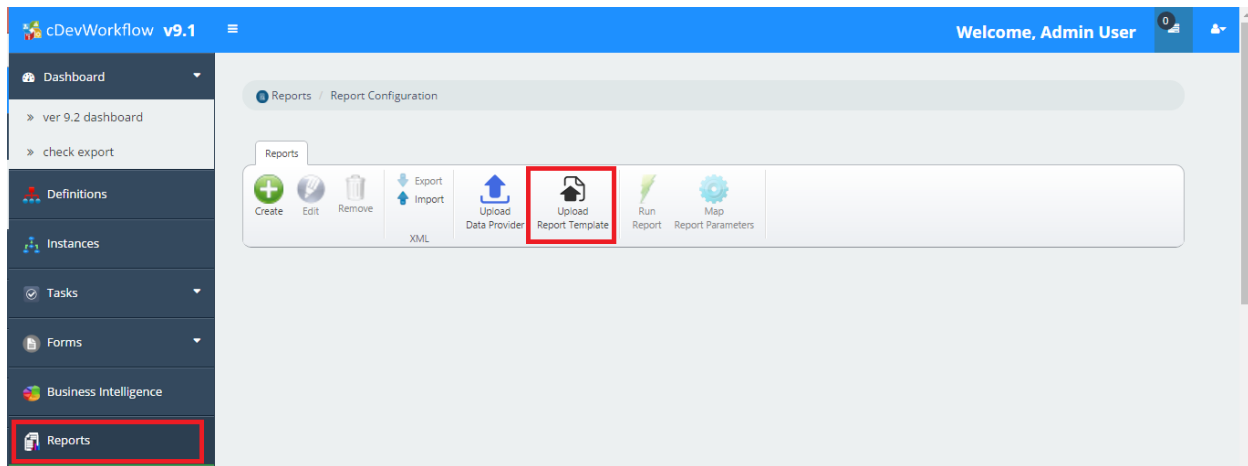
DataSet – StepReport

Parameters – stepName



And, after the SSRS (RDLC) report template is done.

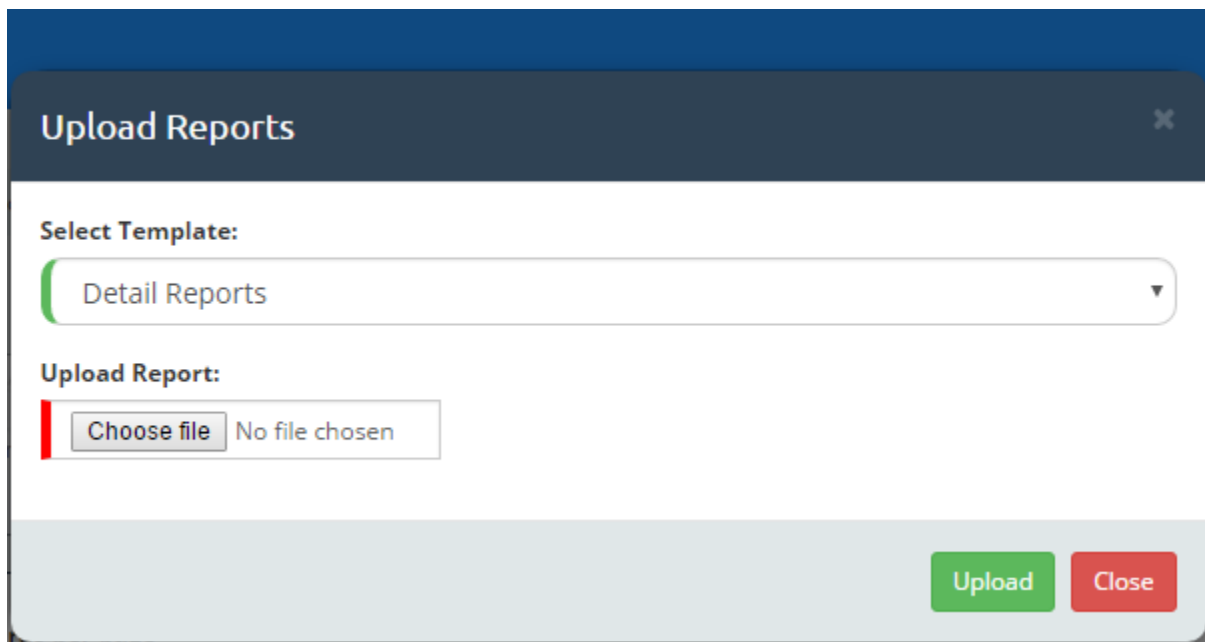
Upload, the template using “Upload Report Template”



When the “Upload Report Template” is clicked select the report category and upload the report template.

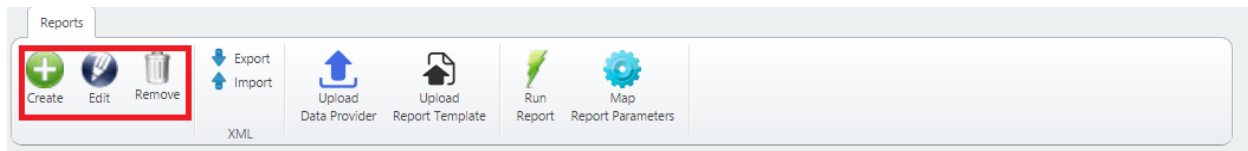
We have added three report category namely (Details Report, Summary Report and User Report)

Select Category, choose the report file and click Upload. The report will be uploaded to “Report Templates” folder in cDevWorkFlow under the selected category (i.e., with same name of the folder).

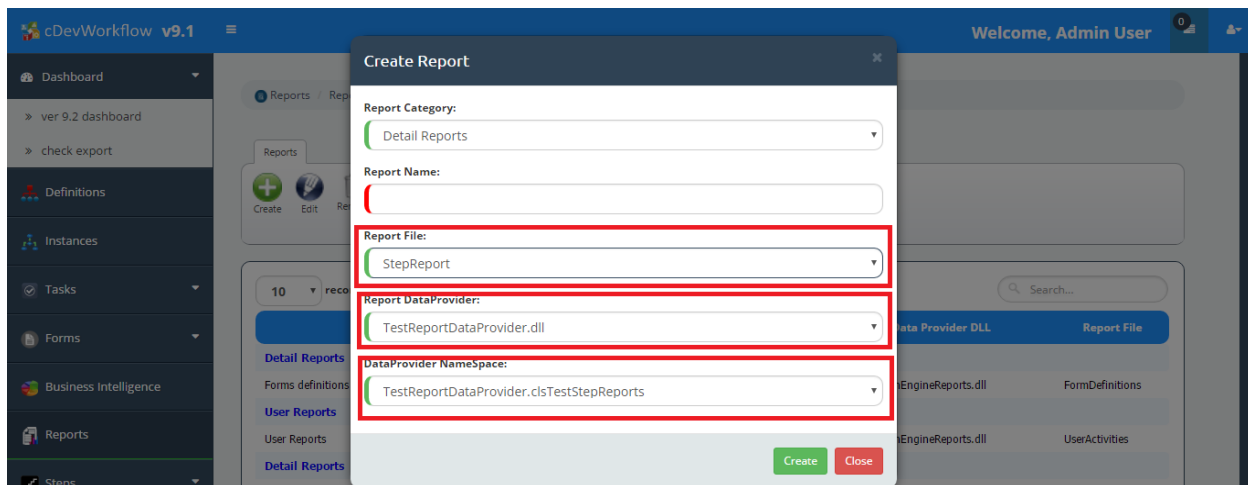


## Mapping between the Report Template and the Data Provider

After Uploading of Data-Provider and Report Template.



Click “Create” icon in the page.



Enter the required details and click “Create” button.

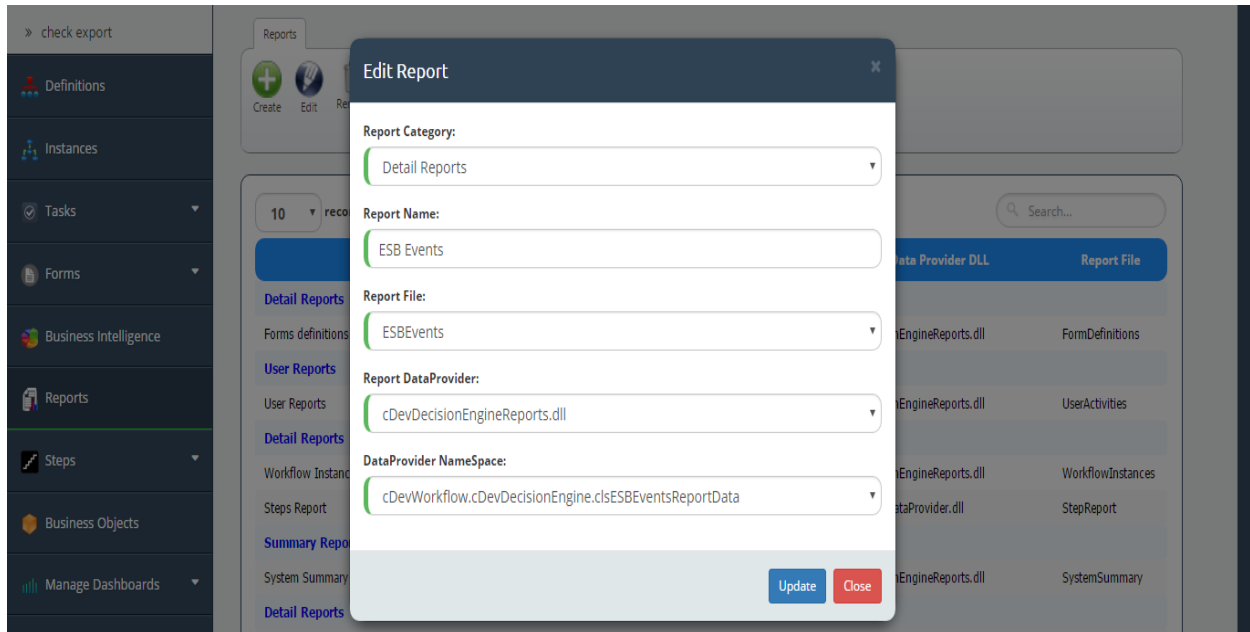
The Uploaded “StepReport” is present under “Detail Reports” Category. The DataProvider is present in “Report DataProvider” Dropdown “TestReportDataProvider.dll”

And, the class name given in the “TestReportDataProvider” is present in

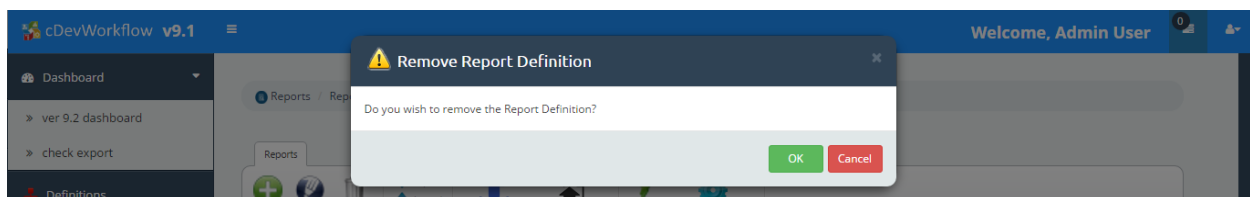
DataProvider Namespace “TestReportDataProvider.clsTestStepReports”

New Report Definition will be generated which can be viewed in the grid.

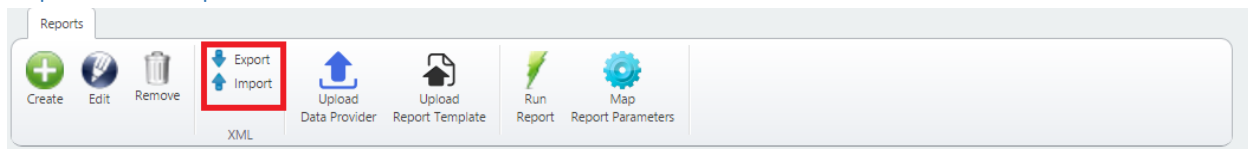
For “Edit”: Select the Report definition on grid and click “Edit” or “Double click the record”



For “Remove”: Select Report definition on grid and click “Remove”. On confirmation the Report definition is deleted or removed.

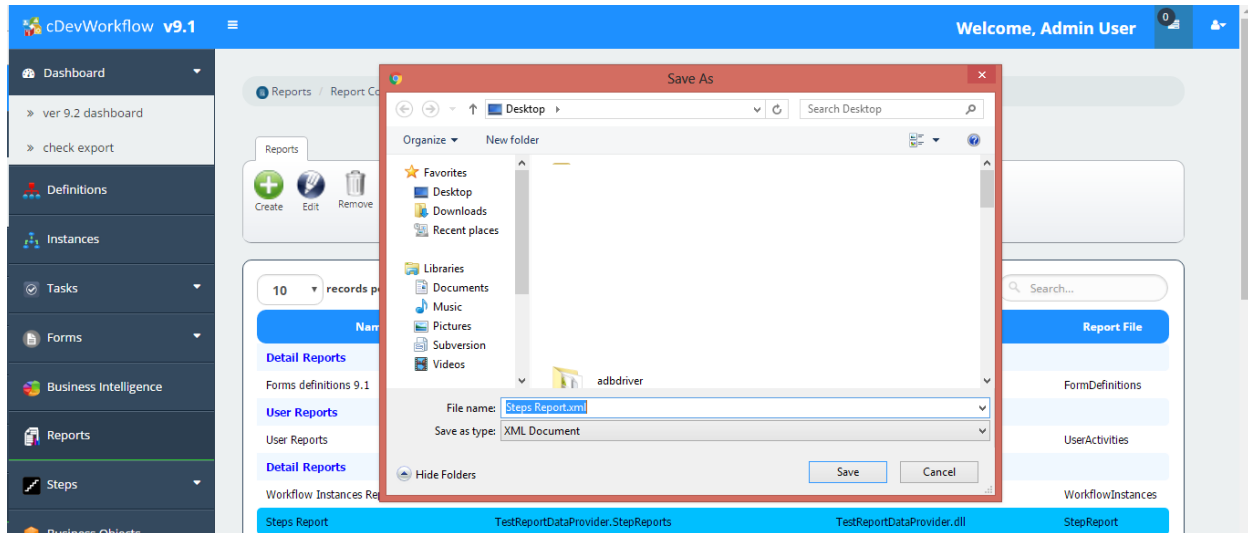


## Export and Import



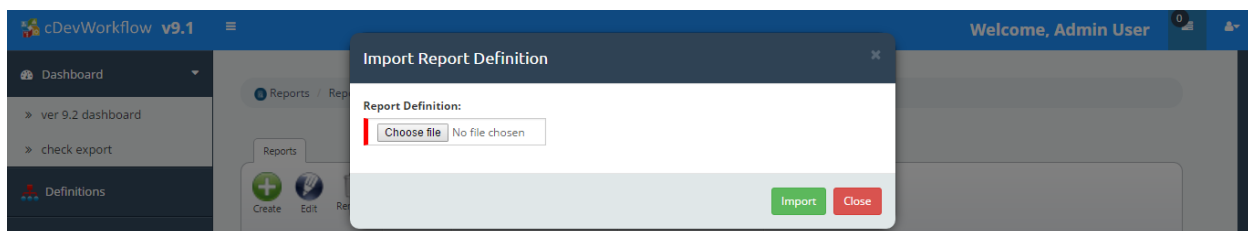
The Report definition can be exported to XML and can be imported from XML.

For “Export”: Select any Report Definition on grid and click “Export”. The Definition will be downloaded as XML file with Report Definition details.

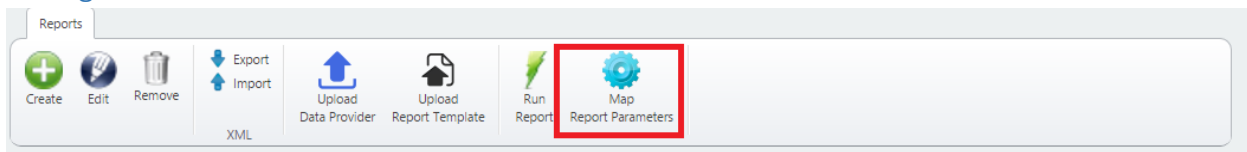


For “Import”: Click “Import” and choose XML file and Click “Import”.

If the Report Name is already present, it updates the report definition. If not then, creates new Report Definition.



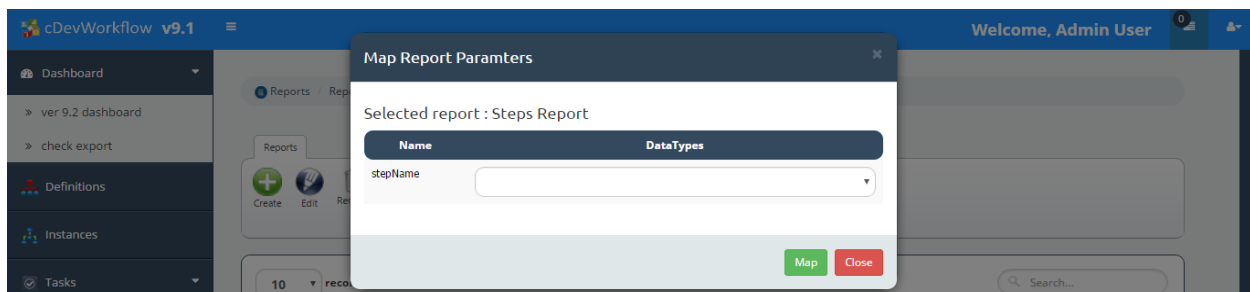
## Configure Parameters



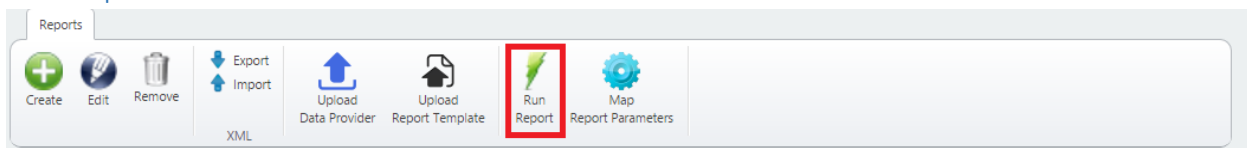
The Report Parameters can be mapped to the DataTypes present in cDevWorkflow. And, the DataTypes will be rendered during “Run Report”.

Select any Report Definition where Parameters mapping required with DataTypes.

**Note: If Report has Parameters and not mapped it'll be rendered as DataType “string”.**



## Run Report

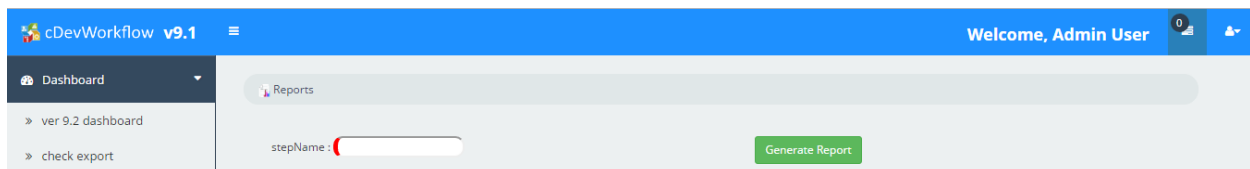


After all the configuration ,

Select any Report Definition required to Run. And, click “Run Report”.

New Tab will open with Parameters Rendered as Mapped DataTypes or string with Button “Generate Report”

In the present sample, the “stepName” will be rendered as “string” DataType.



Enter any Step Name, Relevant Data will be rendered in Report Viewer.

stepName: eval Generate Report

1 of 1

### Steps Report

Step Name	Step Display Name	Step NameSpace	Step DLL
evaluatedecisions	Evaluate decisions	cDevWorkflow.cDevDecisionEngine.evaluateDecisionsStep	cDevDecisionEngineSteps.dll
evalexpression	Evaluates an C# expression and assigns the value to variable	cDevWorkflow.cDevDecisionEngine.evalExpressionStep	cDevDecisionEngineSteps.dll

Report By: cDevWorkflow Testing Date Created: 11/25/2016

The Report can be exported to either Excel or PDF. By clicking on the “Save” Icon.

