Aras Innovator 11

Microsoft Reporting Services Guide

For use with Microsoft SQL Server 2008 R2

Document #: 11.0.02014120801

Last Modified: *12/30/2014*

aras corp®

Copyright Information

Copyright © 2014 Aras Corporation. All Rights Reserved.

Aras Corporation
300 Brickstone Square
Suite 700
Andover, MA 01810

Phone: 978-691-8900
Fax: 978-794-9826

E-mail: Support@aras.com

Website: http://www.aras.com

Notice of Rights

Copyright © 2014 by Aras Corporation. This material may be distributed only subject to the terms and conditions set forth in the Open Publication License, V1.0 or later (the latest version is presently available at http://www.opencontent.org/openpub/).

Distribution of substantively modified versions of this document is prohibited without the explicit permission of the copyright holder.

Distribution of the work or derivative of the work in any standard (paper) book form for commercial purposes is prohibited unless prior permission is obtained from the copyright holder.

Aras Innovator, Aras, and the Aras Corp "A" logo are registered trademarks of Aras Corporation in the United States and other countries.

All other trademarks referenced herein are the property of their respective owners.

Notice of Liability

The information contained in this document is distributed on an "As Is" basis, without warranty of any kind, express or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose or a warranty of non-infringement. Aras shall have no liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this document or by the software or hardware products described herein.

Table of Contents

Send Us Your Comments **5**

Document Conventions **6**

1 Introduction **7**

2 Installing Reporting Services **8**

2.1 Installation is a component of SQL Server 2008 8

2.1.1 Default Instance *9*
2.1.2 Service account Setup *9*
2.1.3 Analysis Services Configuration *10*
2.1.4 Authentication Mode *10*
2.1.5 Reporting Services Options *11*
2.1.6 Remaining Selections *11*

3 Configuring Reporting Services **12**

3.1 Add Reports Folder 12
3.2 Add Shared Data Source 13
3.3 Add Reporting User and Role 15

3.3.1 Add local User Account for Reporting *15*
3.3.2 Add the Reporting Role *16*

4 Using Visual Studio for Report Design **20**

4.1 Create a New Report Project 20
4.2 Configure Project Properties 21
4.3 Add New Shared Data Source 21

4.3.1 Connection Properties *21*
4.3.2 Deploy the Shared Data Source *23*

5 Creating your First Report **24**

5.1 Start a New Report using the Wizard 24
5.2 Select the Data Source 25
5.3 Design the Query 26
5.4 Select the Report Type 27
5.5 Design the Table 28
5.6 Choose the Table Style 29
5.7 Name the Report 30
5.8 Wizard Report Results 31
5.9 Visual Studio Layout 31
5.10 Deploying the Report 32
5.11 Confirm Deployment 32
5.12 Running the Report from within Aras Innovator 32

5.12.1 Adding Aras Innovator Report Item *33*
5.12.2 Passing Named Parameters to the ReportServer *33*

©2014 Aras Corporation All Copyrights Reserved.

6 Report Server query_string Parameters 35

6.1 Named Parameters 35
6.2 Format Parameter 35
6.3 Report Toolbar 35
6.4 Parameters Form 35
6.5 Zoom 35
6.6 Reports Folder 36
6.7 Report Name 36

7 Report Manager 37

7.1 Opening the Report Manager page 37
7.2 Export the RDL 38
7.3 Import the RDL 39
7.4 Select the Shared Data Source 41

8 Graphical Query Builder 43

9 Date Related SQL Queries 45

10 Cookbook 46

10.1 User Input Reports 46
10.2 Report Chooser 48
10.3 Date Selector 49

©2014 Aras Corporation All Copyrights Reserved.

Send Us Your Comments

Aras Corporation welcomes your comments and suggestions on the quality and usefulness of this document. Your input is an important part of the information used for future revisions.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where and what level of detail?
- Are the examples correct? Do you need more examples?
- What features did you like most?

If you find any errors or have any other suggestions for improvement, indicate the document title, and the chapter, section, and page number (if available).

You can send comments to us in the following ways:

Email:
Support@aras.com
Subject: Aras Innovator Documentation

Or,

Postal service:
Aras Corporation
300 Brickstone Square
Suite 700
Andover, MA 01810
Attention: Aras Innovator Documentation

Or,

FAX:
978-794-9826
Attn: Aras Innovator Documentation

If you would like a reply, provide your name, email address, address, and telephone number.

If you have usage issues with the software, visit http://www.aras.com/support/

©2014 Aras Corporation All Copyrights Reserved.

Document Conventions

The following table highlights the document conventions used in the document:

Table 1: Document Conventions

Convention	Description
Bold	This shows the names of menu items, dialog boxes, dialog box elements, and commands. Example: Click **OK**.
`Code`	Code examples appear in `courier` text. It may represent text you type or data you read.
`Yellow highlight`	Code with yellow highlight is used to draw attention to the code that is being indicated in the content.
`Yellow highlight with red text`	Red color text with yellow highlight is used to indicate the code parameter that needs to be changed or replaced.
Italics	Reference to other documents.
Note:	Notes contain additional useful information.
Warning	Warning contains important information. Pay special attention to information highlighted this way.
Successive menu choices	Successive menu choices may appear with a greater than sign (-->) between the items that you will select consecutively. Example: Navigate to **File** --> **Save** --> **OK**.

©2014 Aras Corporation All Copyrights Reserved.

1 Introduction

Aras Innovator's Report ItemType includes support for Microsoft's 'Reporting Services' in addition to supporting the existing XSLT based reports.

Reporting Services is an integral component of SQL Server 2008, and any customer with a SQL Server 2008 license can install Reporting Services. (For more information, refer to section 2 Installing Reporting Services).

Usage of this release is recommended only for those with advanced Aras Innovator and SQL training.

Primary benefits to switching from stylesheet driven reports to Report Server:

- Query building is in SQL, and is graphical (same as in Enterprise Manager).
- Report definitions are transportable, you can develop on your laptop, and then deploy to a production server.
- Report layout is more drag-n-drop and has more powerful formatting tools.
- The resulting reports are paginated.
- The end-user can easily print, PDF, or export to Excel the report output.
- Excel like Charts and graphs can be produced

Planned enhancements are a tighter integration of the authentication / permissions model, and finding an alternative to Visual Studio for end-use report layout.

To learn more about Microsoft Reporting Services visit: Microsoft SQL Server: Reporting

©2014 Aras Corporation All Copyrights Reserved.

2 Installing Reporting Services

Reporting Services is a component of Microsoft SQL server 2008 and can be installed by selecting the 'Reporting Services' option during the SQL server installation.

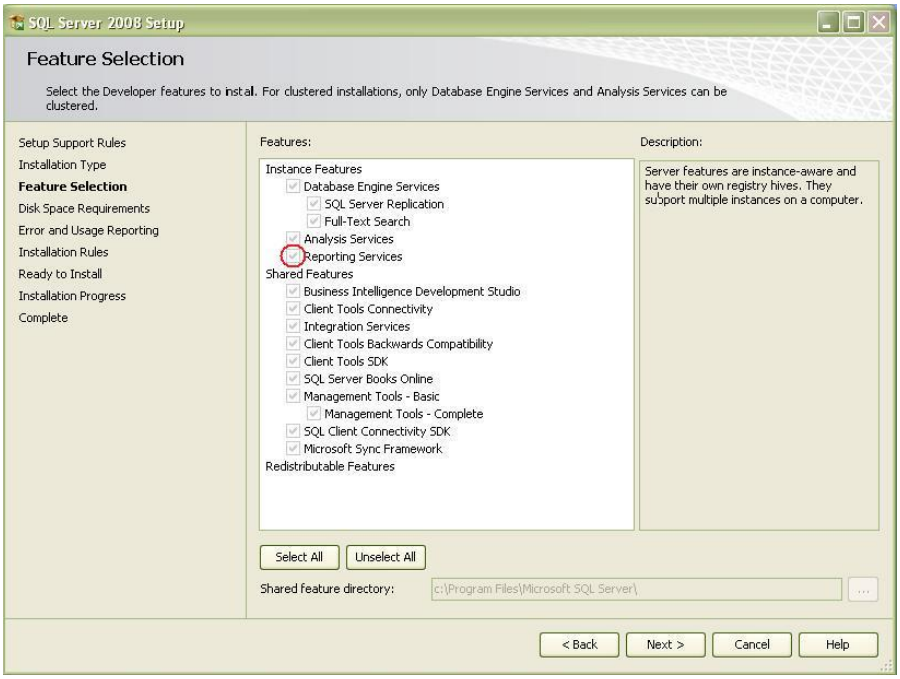


Figure 1.

In addition to selecting the Reporting Services option, you can select 'Analysis Services' which provides graphing and charting functionality.

Selecting the 'Reporting Services' option also installs components of Visual Studio 2008 for report design and deployment

2.1 Installation is a component of SQL Server 2008

The following is a summary of dialogs to be used when installing SQL server 2008.

Note: If you view dialog boxes not listed here, use the default selections.

©2014 Aras Corporation All Copyrights Reserved.

2.1.1 Default Instance

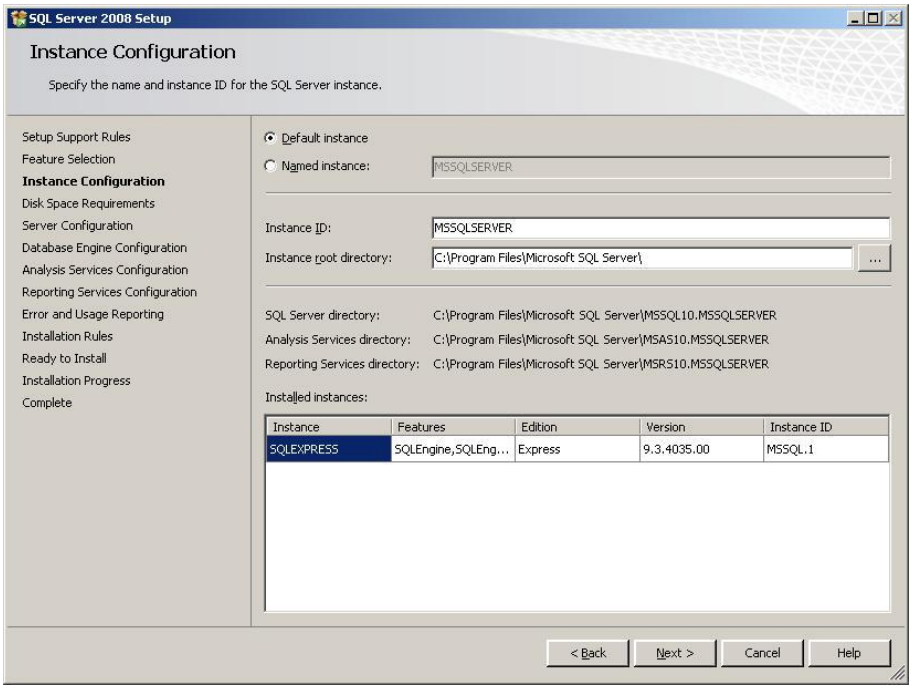


Figure 2.

2.1.2 Service account Setup

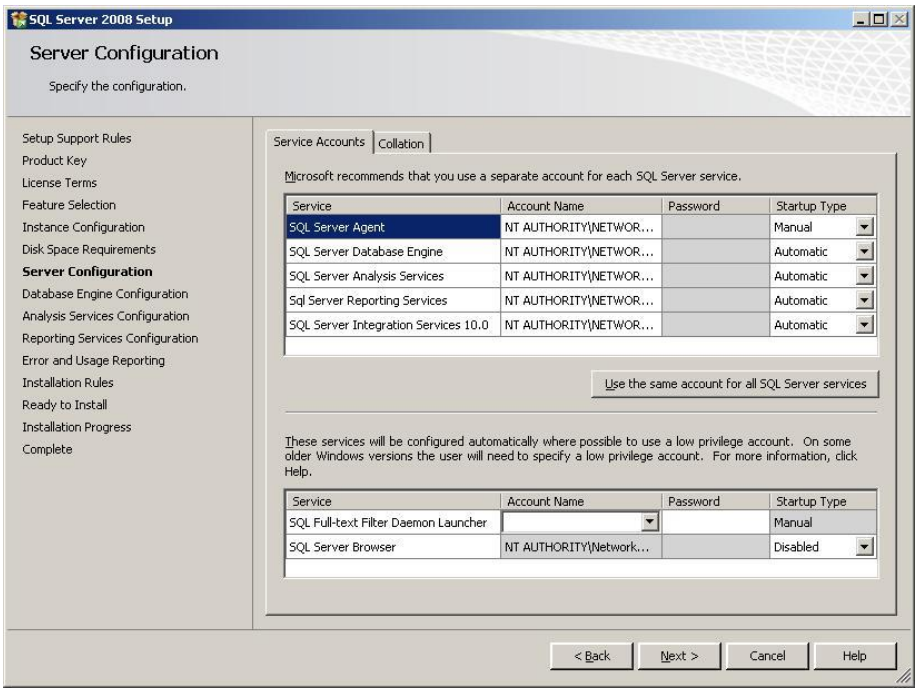


Figure 3.

©2014 Aras Corporation All Copyrights Reserved.

2.1.3 Analysis Services Configuration

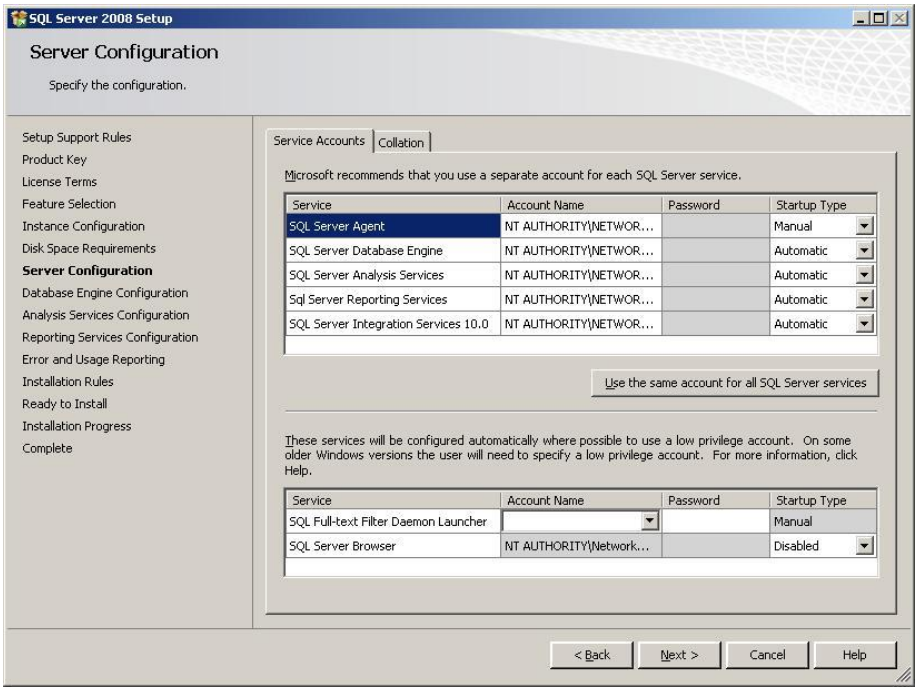


Figure 4.

2.1.4 Authentication Mode

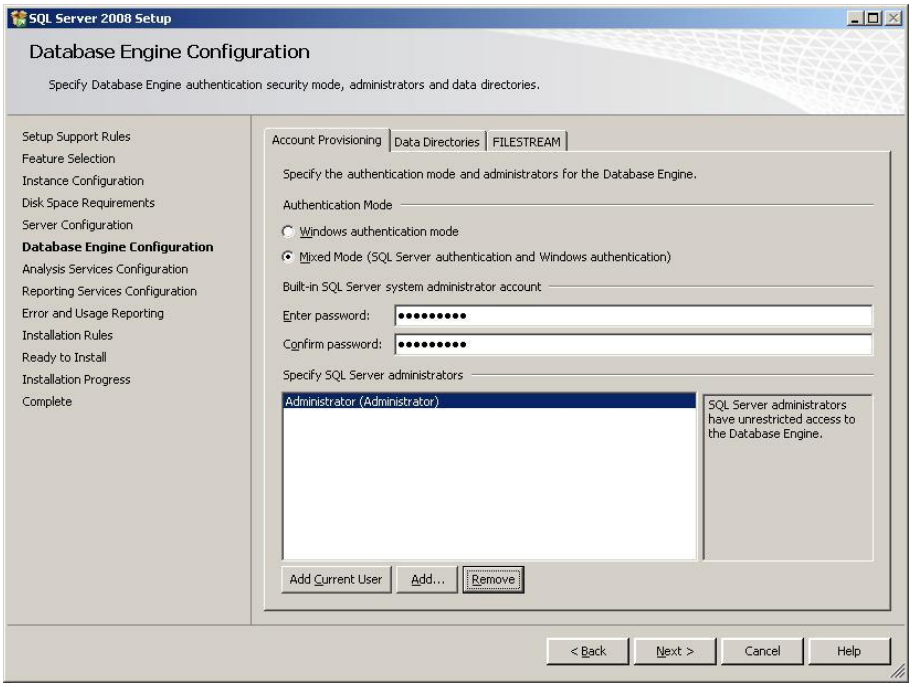


Figure 5.

©2014 Aras Corporation All Copyrights Reserved.

2.1.5 Reporting Services Options

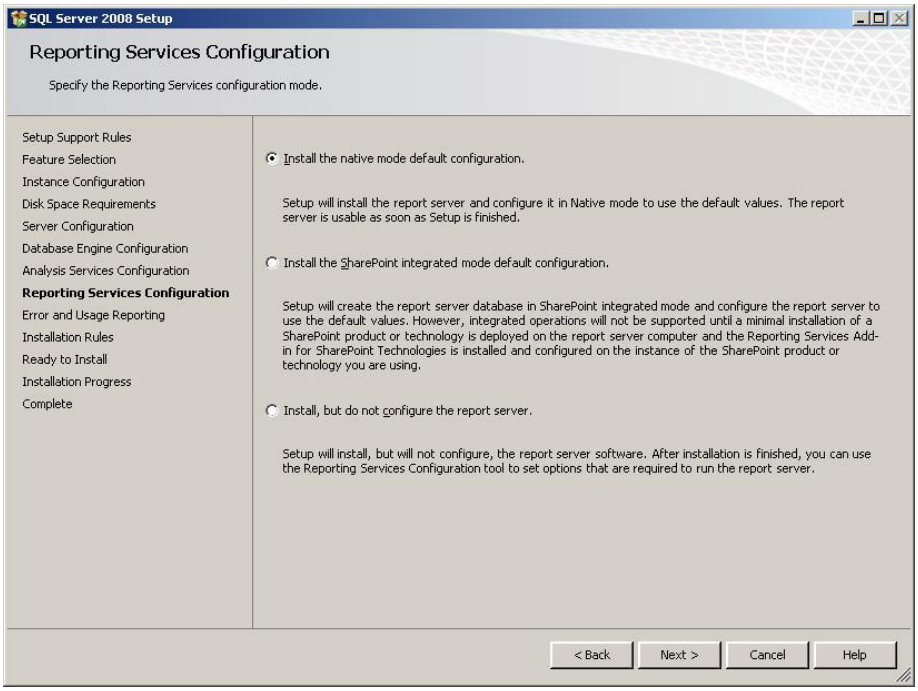


Figure 6.

2.1.6 Remaining Selections

The default selections can be used from this point forward.

©2014 Aras Corporation All Copyrights Reserved.

3 Configuring Reporting Services

The Reporting Services use Folders to organize Reports. The Aras Innovator integration with Reporting Services use as a default the Folder named 'Innovator Reports'.

In this section, we configure the Reporting Services using the Report Manager by adding the 'Innovator Reports' Folder and a Shared Data Source, which the Reports use to connect to the database.

3.1 Add Reports Folder

The Reporting Services ReportServer comes with a Report Manager page used to configure the Reports, Folders, Data Sources, and so on. This page has been integrated into Innovator client via an Innovator Report that opens the Report Manager page.

Open your browser with the URL: http://localhost:8080/Reports (on a default install)

If your installation is not default, open the Reporting Services Configuration Manger to verify the URL.

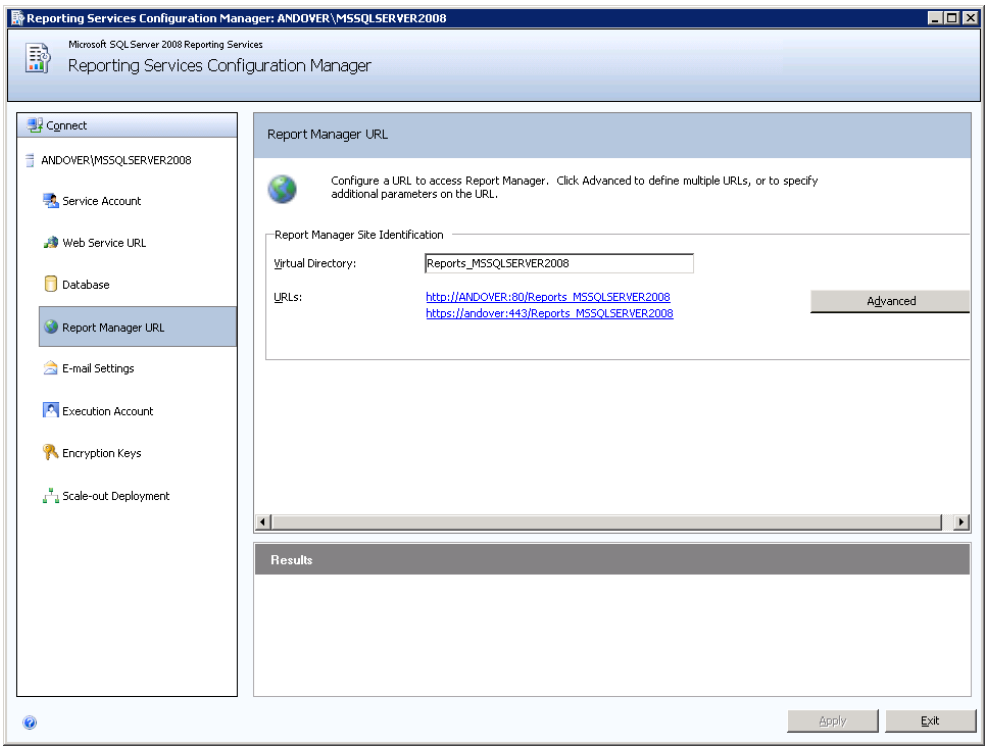


Figure 7.

©2014 Aras Corporation All Copyrights Reserved.

1. The **Report Manager** page opens in a new window as shown below:

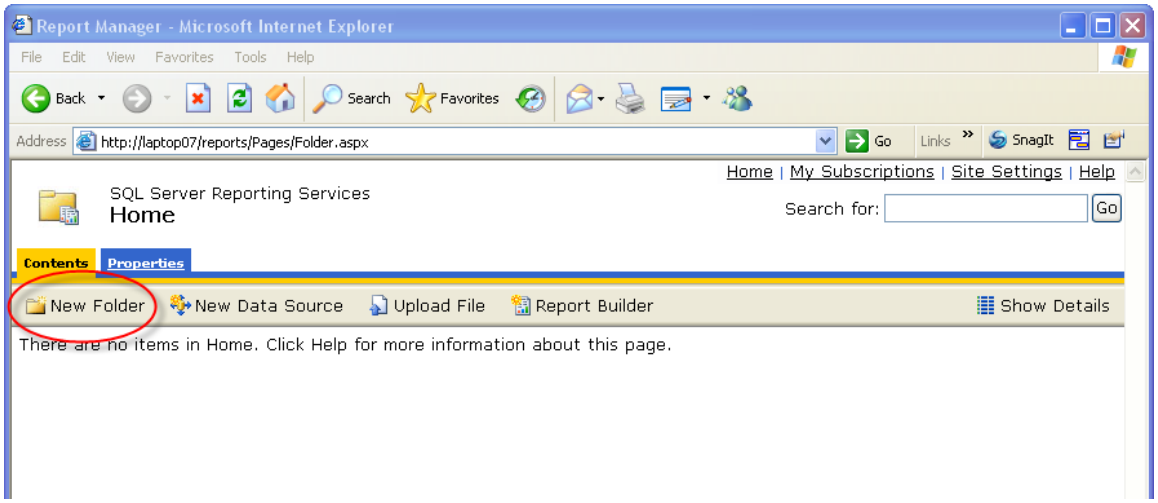


Figure 8.

2. Click **New Folder**.
3. Enter '**Innovator Reports**' for the name and click **OK**.

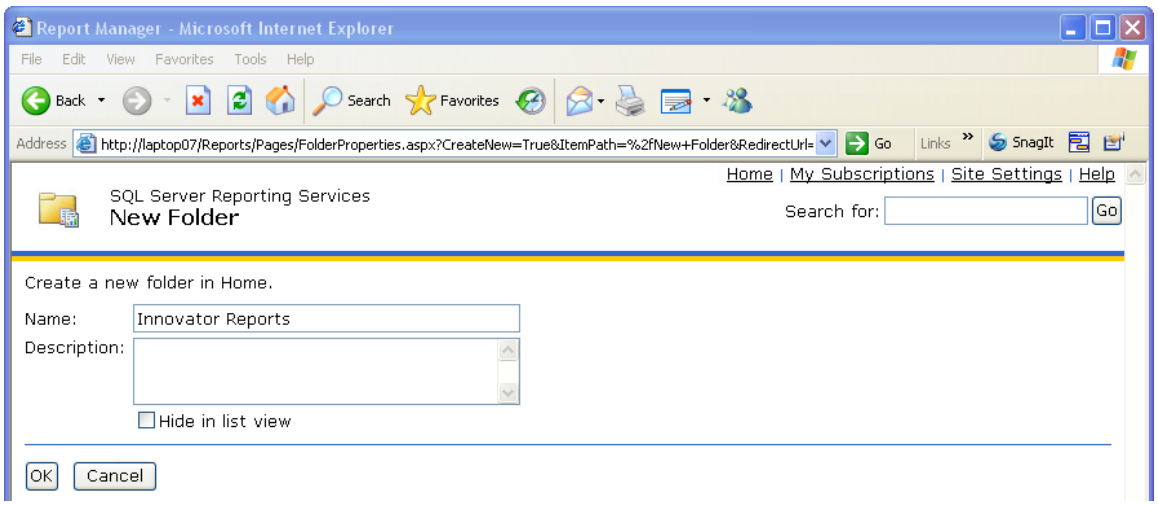


Figure 9.

If you use a folder named something other than 'Innovator Reports', you need to change the 'Report Server Folder' variable in the database as well in Administration\Variables in the TOC. The default value for this is 'Innovator Reports' but can be customized for each database. For the purposes of these instructions, we assume the default value is used.

3.2 Add Shared Data Source

Once you add the 'Innovator Reports' Folder, navigate into it by clicking on the link. Now, add a Shared Data Source for the Reports to use to connect to the database.

©2014 Aras Corporation All Copyrights Reserved.

1. Click **New Data Source** in the center of the page.

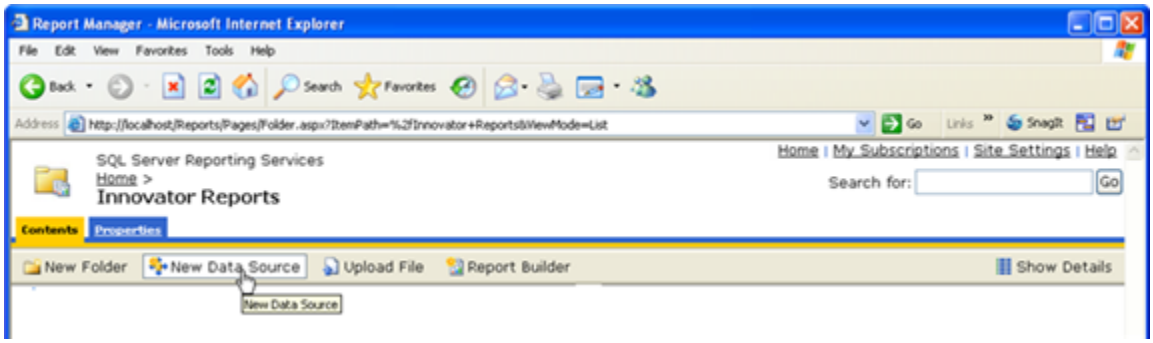


Figure 10.

2. Enter '**Innovator**' for the **Data Source Name**. By using this same name for both the development and production servers exporting/importing Reports becomes simpler because there is no need to reassign the Data Source for the Reports you import.
3. Enter the following for the 'Connection String' replacing the values for the two parameters with the actual server name where SQL Server is installed and the actual database name.

```
data source=<SQLServerName>;initial catalog=<InnovatorDatabase>
e.g.
data source=server1;initial catalog=innovator
```

4. Select the '**Credentials stored securely in the report server**' option for the '**Connect Using**' Field and enter the login credentials for your database.
5. Click **Apply** to deploy the shared Data Source.

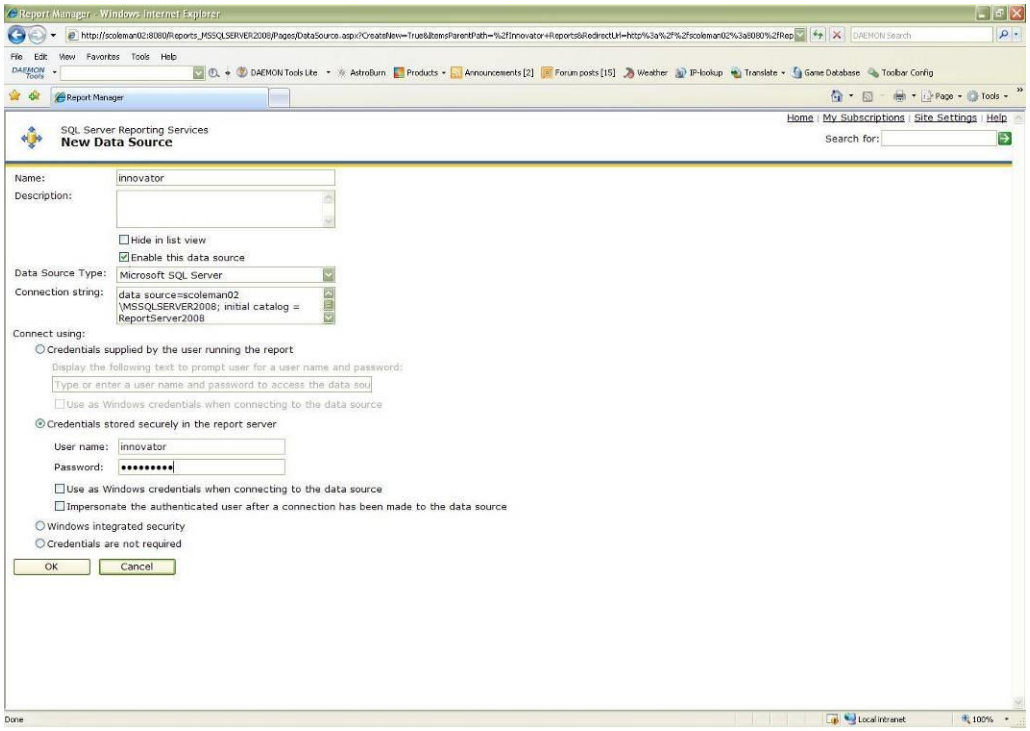


Figure 11.

©2014 Aras Corporation All Copyrights Reserved.

3.3 Add Reporting User and Role

The ReportServer.asmx Web Service is the Report Server entry point for the Microsoft Reporting Services. The same Web Service is used to request Reports and administer the Report Server. The Web Service is by default configured to use Windows Authentication. You can't simply open the security for anonymous Web users because that would open the Report Manager as well.

This security model requires the User be authenticated when they request Reports. A dialog opens requesting the user's login credentials the first time they request a Report each time the browser is opened again. When requesting the Report from within Aras Innovator this would not be desirable. This also means the user must have a domain or local User Account on the server and that becomes an issue for not 100% Microsoft shops with Intranet only Aras Innovator.

The initial release of the Aras Innovator Reporting Services uses one shared User Account for all Report requests. The Aras Innovator Client calls the Aras Innovator Reporting Services Gateway to request Reports. The Gateway submits the actual Report request using this shared User Account and propagates the results back to the client. The Report renders without the authentication dialog appearing.

If the user attempts to view the Report directly outside Aras Innovator they get the authentication dialog as normal and unless the IT department shares the credentials, the users can only view the Reports through Aras Innovator.

The local User Account on the server by default is named ‘Reporting’ with a default password is ‘reporting’ and must be mapped as a Role in the Report Server.

3.3.1 Add local User Account for Reporting

The administrator for the server where Microsoft Reporting Services is installed must add a new Windows User Account named ‘Reporting’. Once the User Account is created you can configure Aras Innovator by setting the user name and password in the InnovatorServerConfig.xml configuration file:

```
<ReportingServices>
  <ReportServer>http://localhost/ReportServer/</ReportServer>
  <User>Reporting</User>
<Password>reporting</Password>
<Domain>Domain for the above User</Domain>
</ReportingServices>
```

Optional node, <`Domain`>, should contain the domain name you want to use to connect to Reporting services.

©2014 Aras Corporation All Copyrights Reserved.

3.3.2 Add the Reporting Role

The Reporting User Account must next be mapped as a Role in the Reporting Services enabling this account Browser privileges.

1. Click the **Site Settings** link in the upper right hand corner of the Report Manager page.

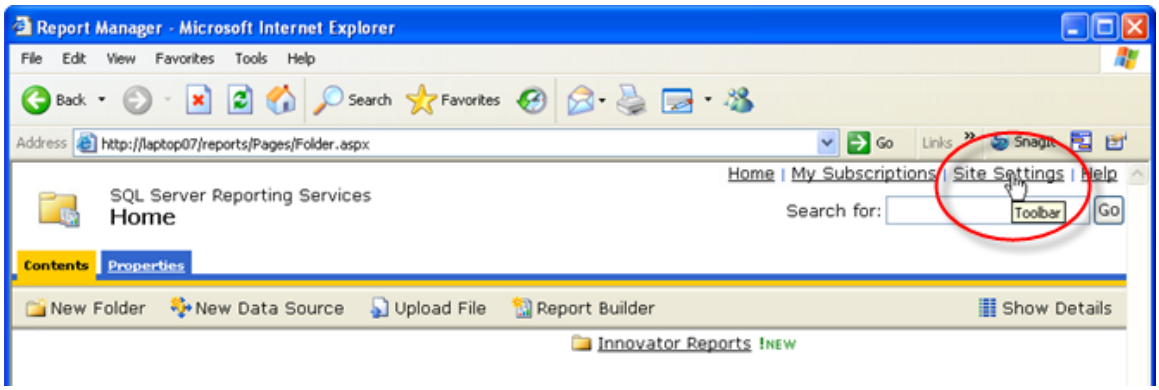


Figure 12.

2. Click the **Configure site-wide security** link.

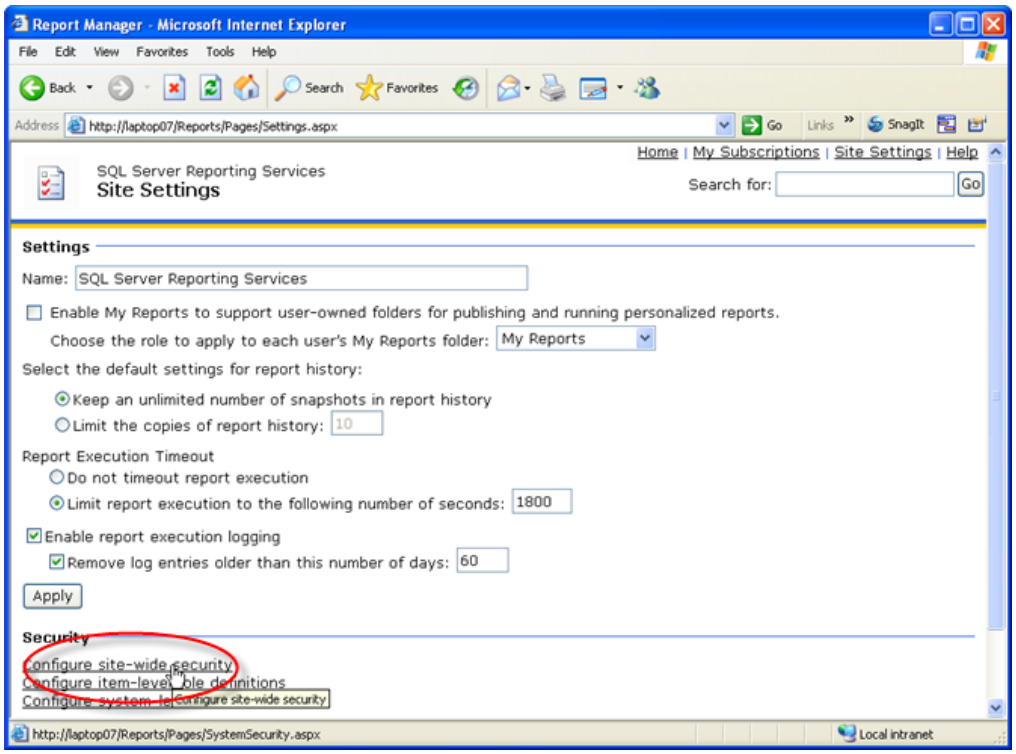


Figure 13.

3. Click **New Role Assignments** to map the Reporting User Account as a Reporting Services Role.

©2014 Aras Corporation All Copyrights Reserved.

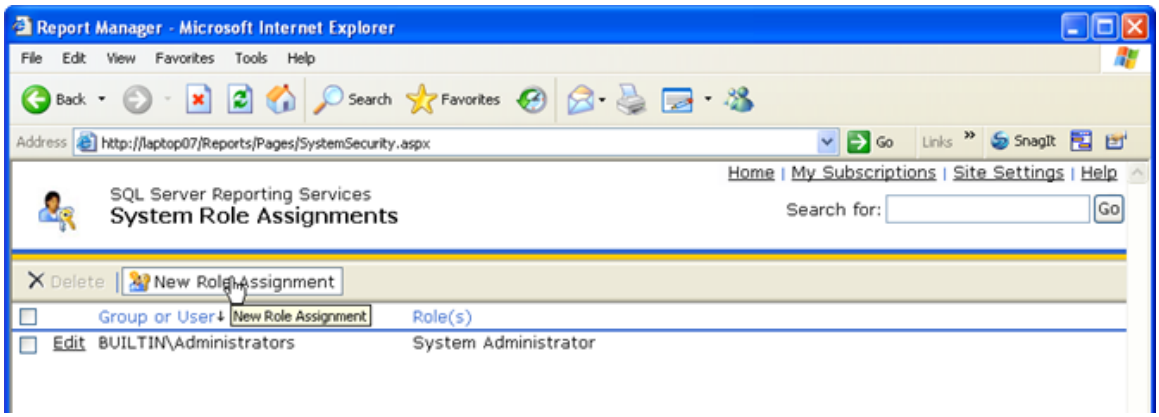


Figure 14.

4. Enter the User Account name Reporting, select the **System User** checkbox, and click **OK** to create the new Role.

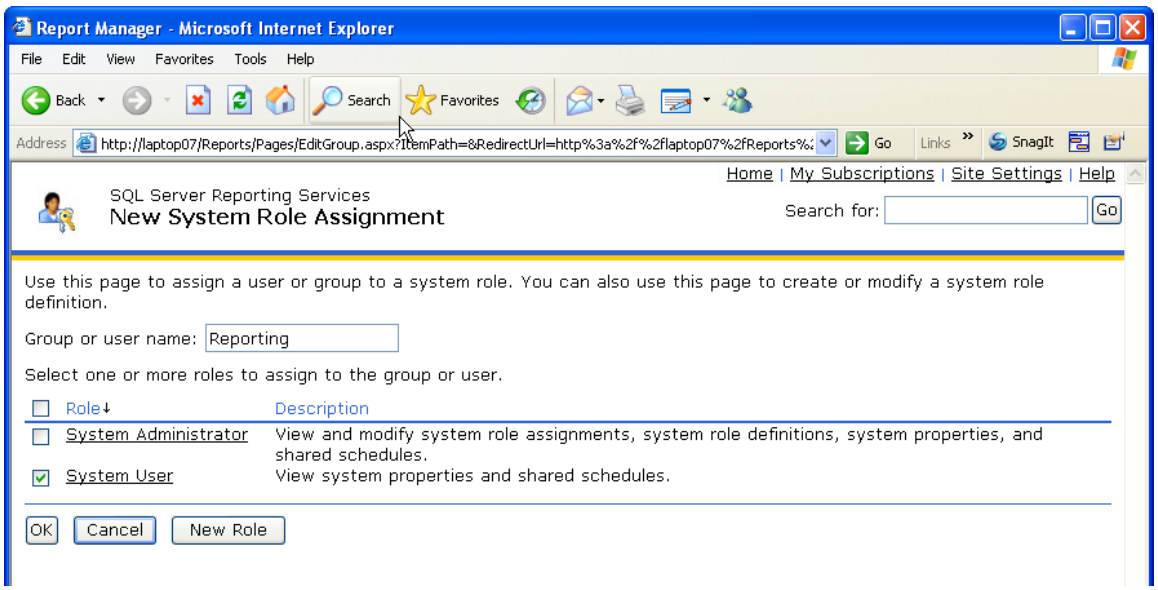


Figure 15.

5. Add the new Reporting Role to the 'Innovator Reports' folder's security. Click the **Home** link in the upper right hand corner on the page, click **Show Details** and select the Edit icon for the Innovator folder.

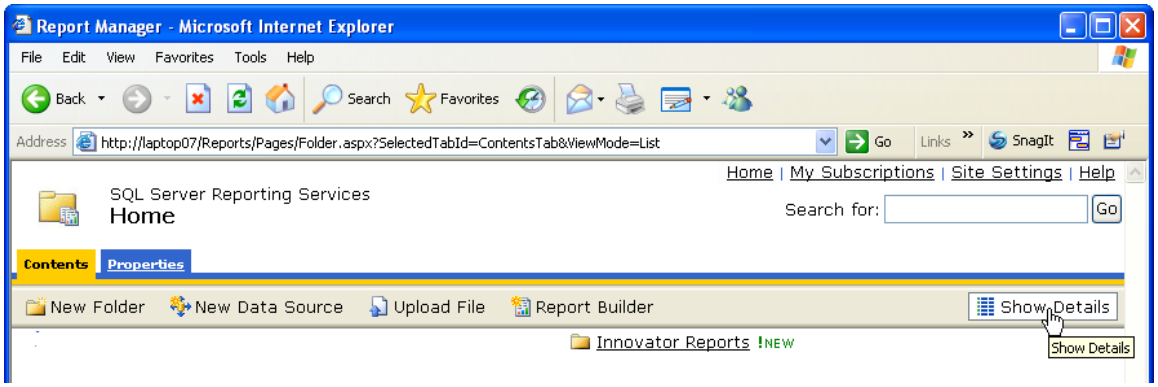


Figure 16.

©2014 Aras Corporation All Copyrights Reserved.

6. Click the **Security** link on the left had side of the page to open the Security page for the Innovator Reports folder.

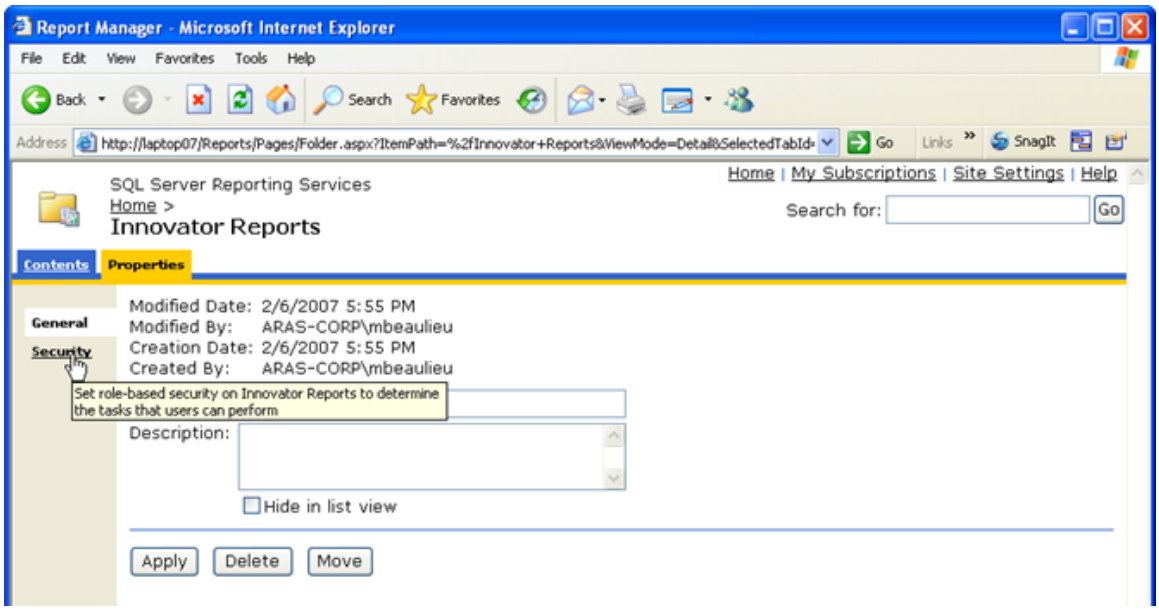


Figure 17.

7. Click **Edit Item Security**.

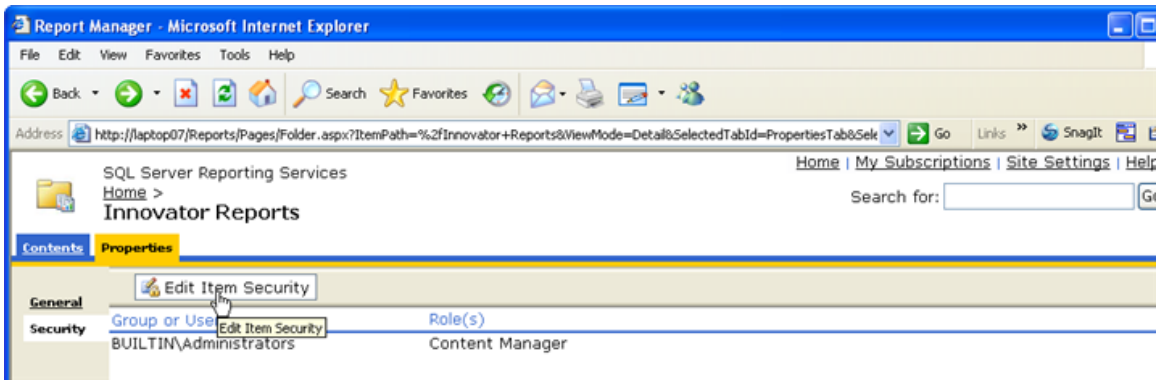


Figure 18.

8. Enter the Reporting Role, check the **Browser** checkbox, and click **OK** to create a new Role Assignment for the Innovator Reports folder.

©2014 Aras Corporation All Copyrights Reserved.

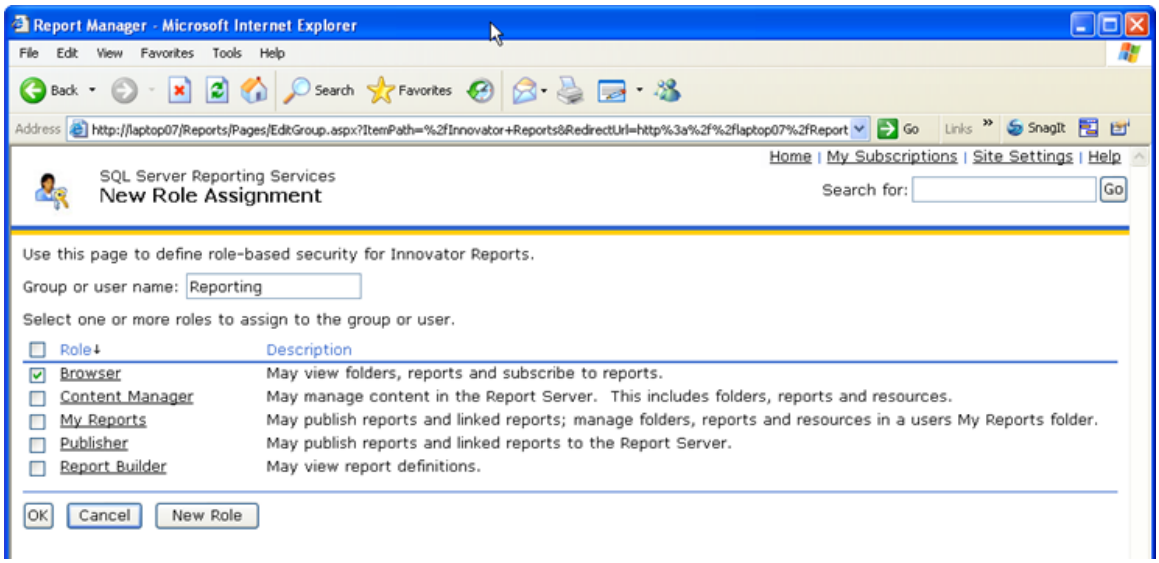


Figure 19.

©2014 Aras Corporation All Copyrights Reserved.

4 Using Visual Studio for Report Design

Once the Reporting Services is installed, you need to configure Visual Studio to create a Project to hold the Reports. Plus configure a Shared Data Source, which is the database connection information for the Reports.

Note: It is required that you follow the exact naming conventions as shown here to ensure that the Aras Innovator integration with the Reporting Services works correctly.

4.1 Create a New Report Project

Start Microsoft SQL Server Business Intelligence Development Studio and create a new project named 'Innovator Reports'. This becomes the Reporting Services Folder to hold the Reports for Aras Innovator.

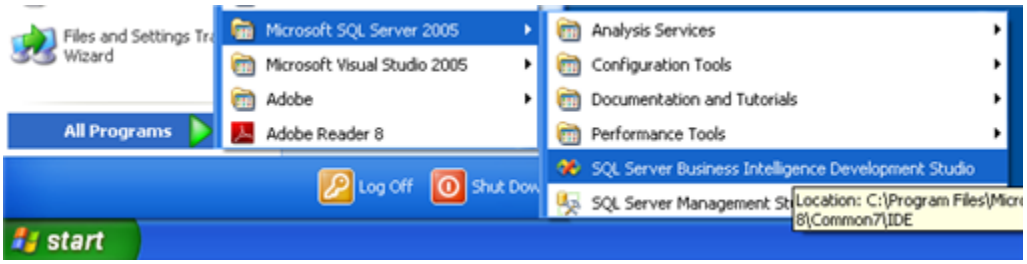


Figure 20.

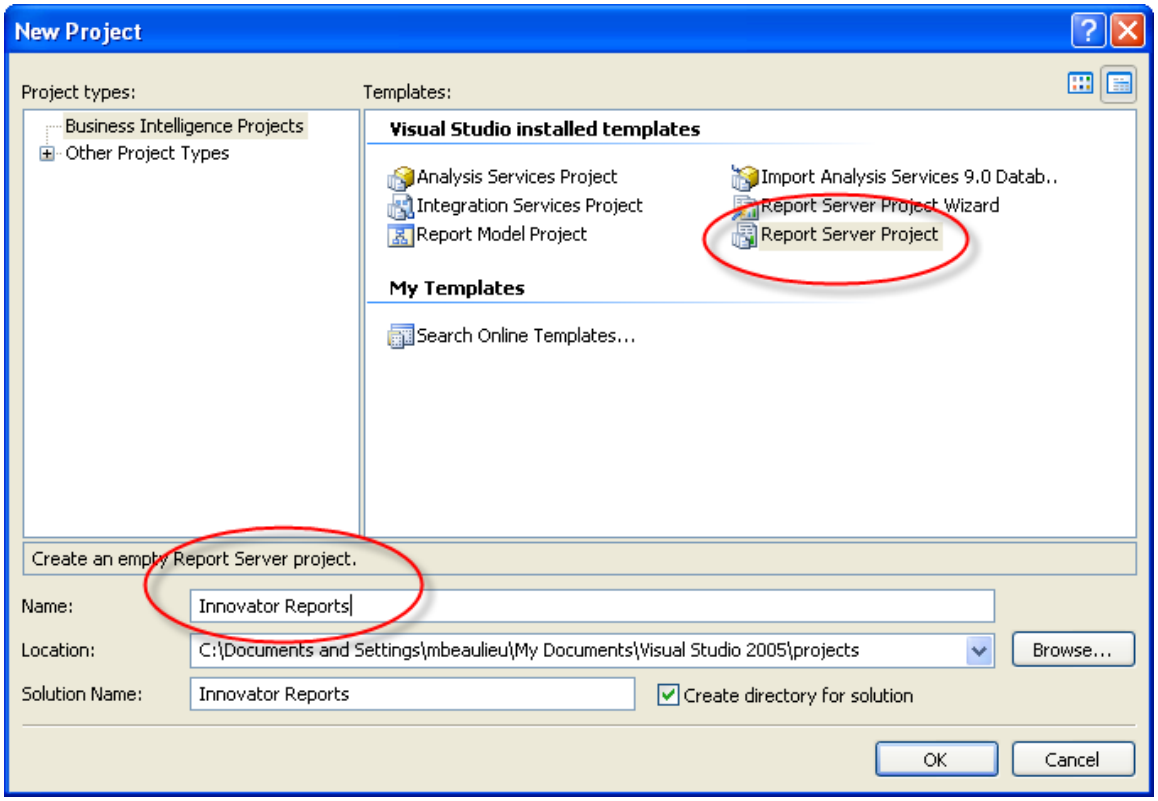


Figure 21.

©2014 Aras Corporation All Copyrights Reserved.

4.2 Configure Project Properties

1. Open the Solutions Explorer panel from the menu **View -->Solutions Explorer**, if it is not already opened.
2. Right-click the **Innovator Reports** folder and select **Properties**.
3. On the Properties dialog, set the TargetServerURl field to the URL of the Report Server.

   For example, http://localhost:8080/ReportServer
   Where, the actual name of the server host name is entered in place of localhost.
   Also, set the OverwriteDateSources flag to True.

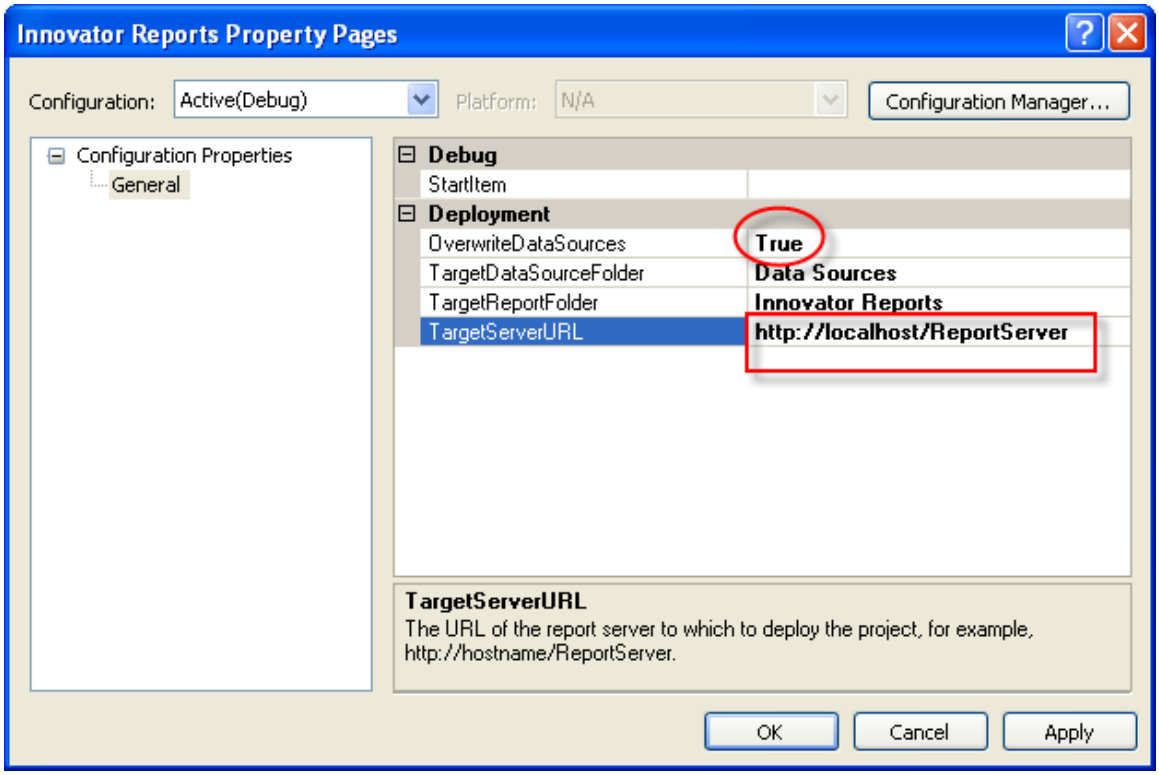


Figure 22.

4.3 Add New Shared Data Source

In the same Solutions Explorer panel, right-click the **Shared Data Sources** folder and select **Add New Data Source**. Click **Edit** to configure the new datasource.

4.3.1 Connection Properties

1. Select or enter the 'Sever name' for the machine SQL Server is installed.
2. Click to select **Use a specific user name and password** and enter the login credentials for the database and check the Save My password.
3. Select or enter the Database name.

©2014 Aras Corporation All Copyrights Reserved.

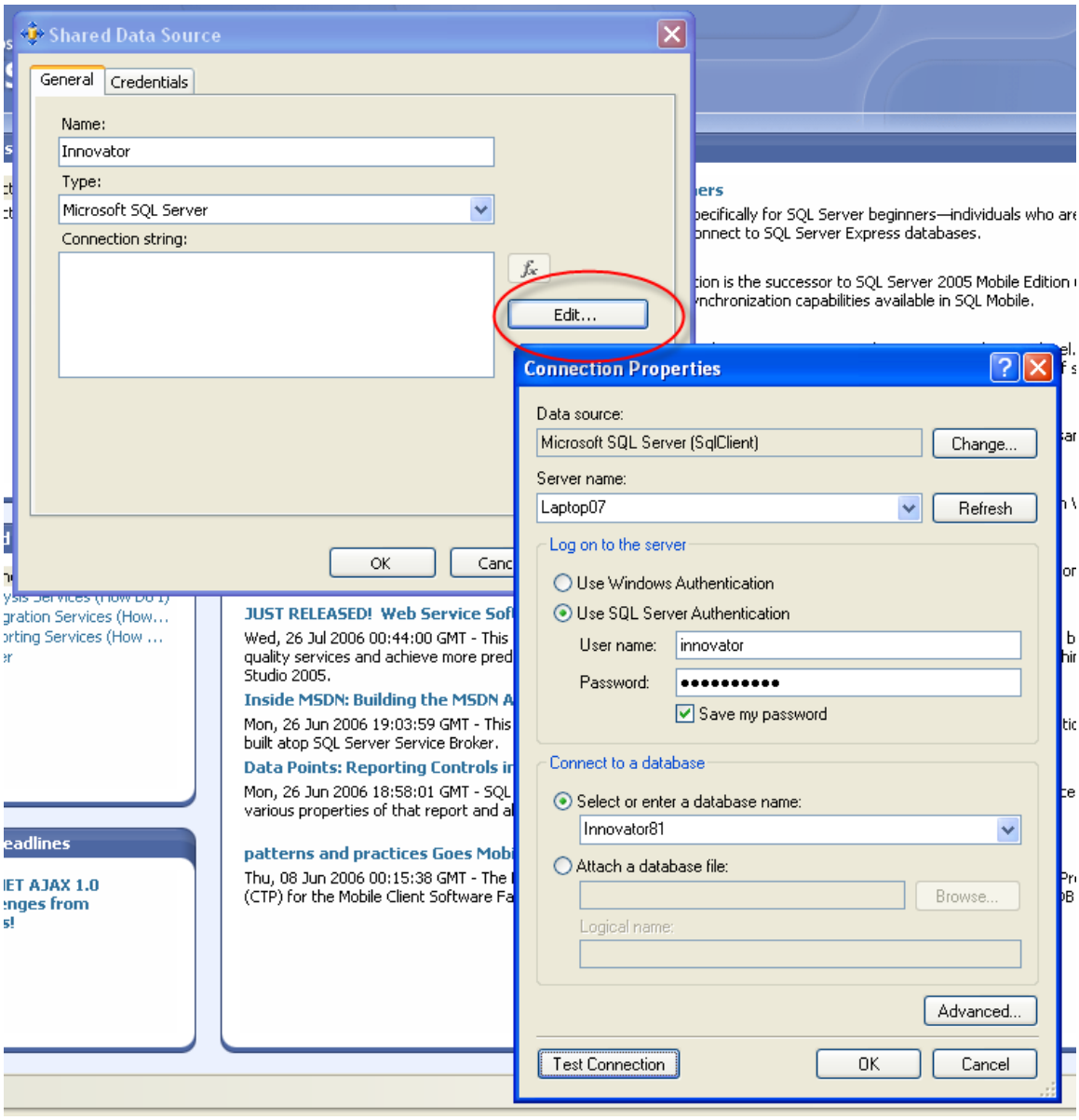


Figure 23.

©2014 Aras Corporation All Copyrights Reserved.

4.3.2 Deploy the Shared Data Source

By default, the name of the Shared Data Source is the name of the Database you selected above. You can rename this if you want. Finally, you need to deploy this new Shared Data Source.

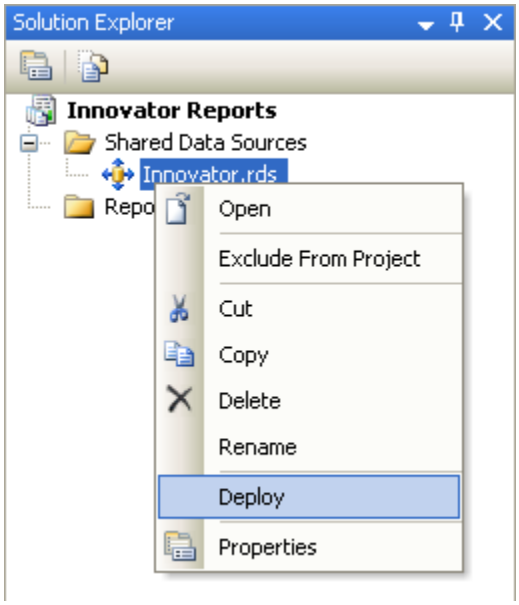


Figure 24.

©2014 Aras Corporation All Copyrights Reserved.

5 Creating your First Report

The following illustrates using the Report Wizard to create a new Report. For this Report, we create a simple User Directory Report showing the Users First Name, Last Name, Email, Phone Number, and Company Name. The following is a screen shot of the Report that we will be creating:

5.1 Start a New Report using the Wizard

From the 'Solution Explorer' panel, right-click the 'Reports' folder, and select **Add New Report**.

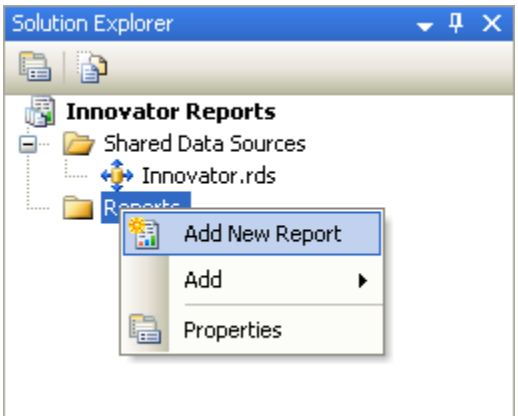


Figure 25.

©2014 Aras Corporation All Copyrights Reserved.

5.2 Select the Data Source

You can create a unique Data Source for this Report, but it is typical that you use the default Shared Data Source that you created.

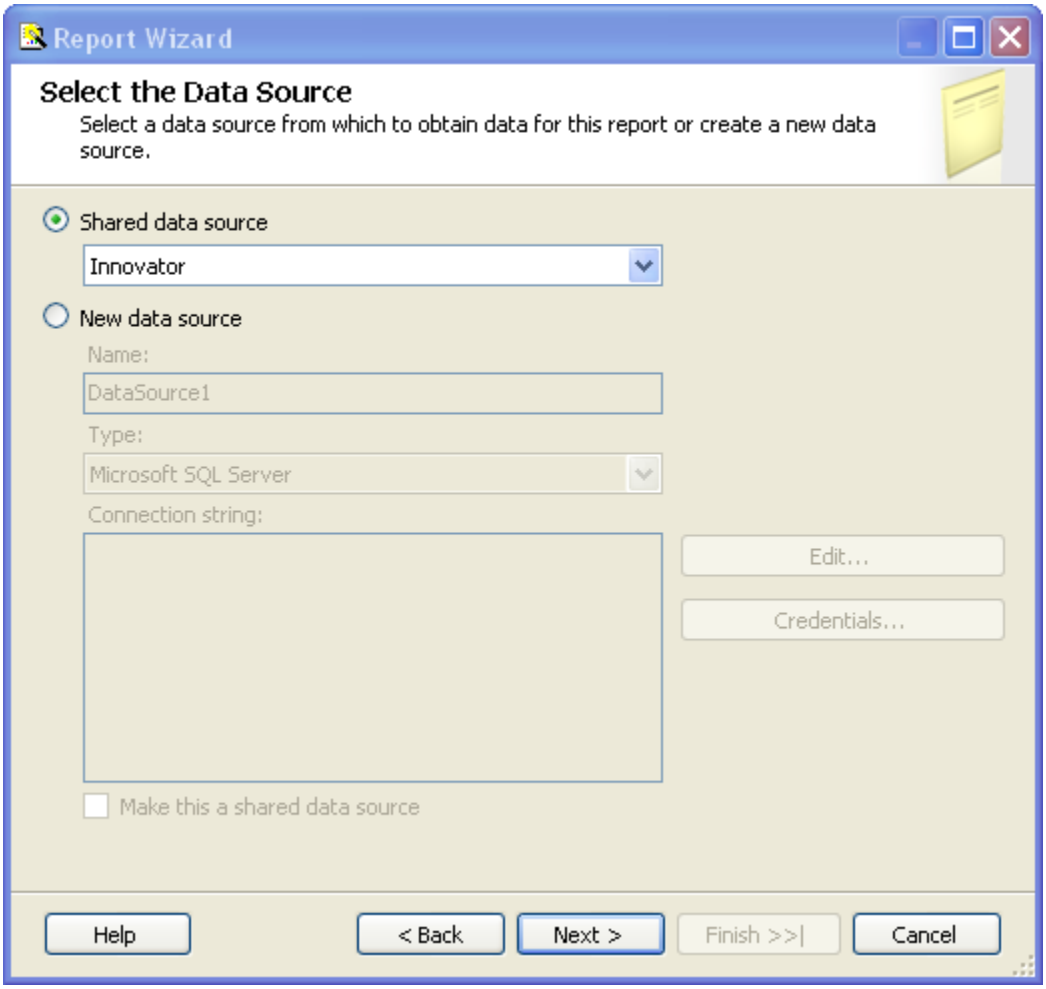


Figure 26.

©2014 Aras Corporation All Copyrights Reserved.

5.3 Design the Query

You need to provide the SQL Select query that fetches the data for the Report. In many cases you can enter the simple select clause in the dialog directly.

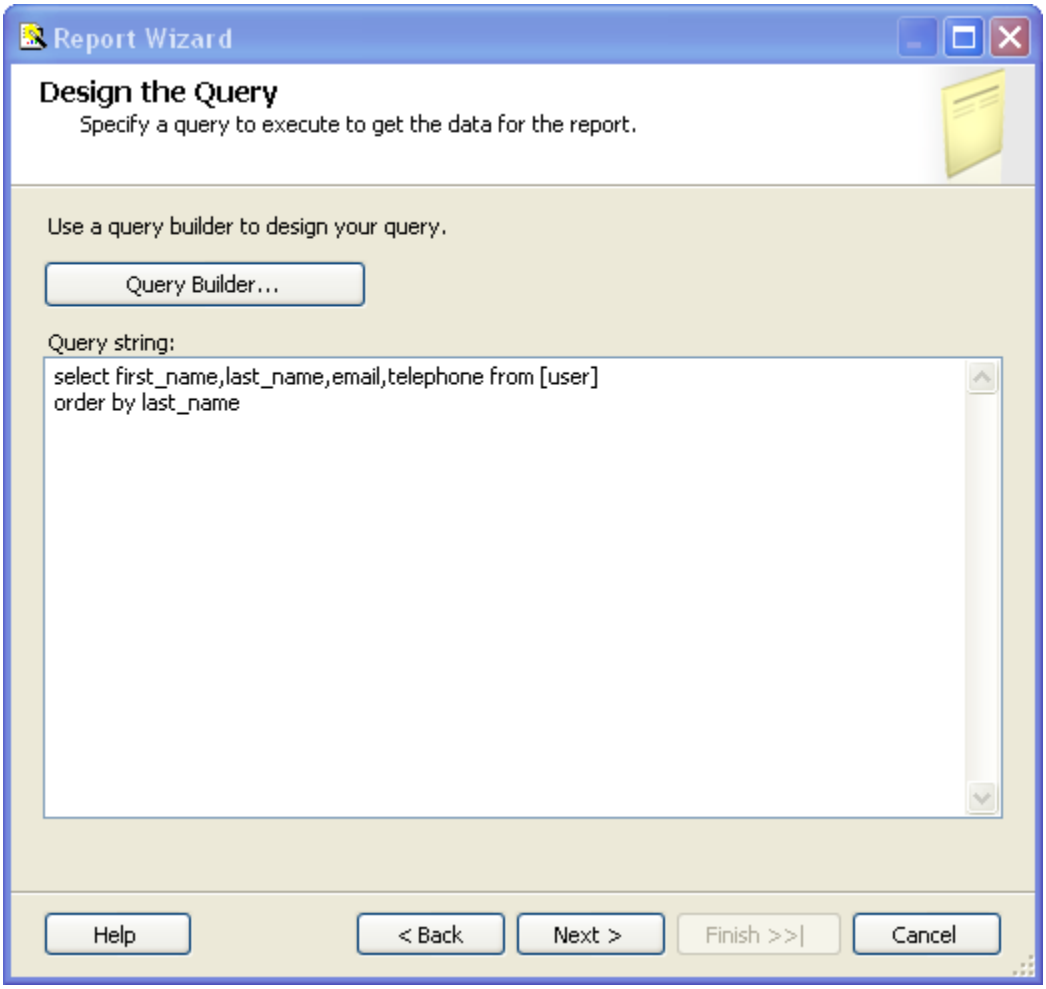


Figure 27.

Note: It is possible to pass Named Parameters into the query such as:
`SELECT * FROM [PROJECT] WHERE id=@ID`
The 'Edit…' button opens a Query Builder interface, which is covered later in more detail.

©2014 Aras Corporation All Copyrights Reserved.

5.4 Select the Report Type

There are two choices for the Report Type: Tabular (column layout) and Matrix (rows and columns layout). Select the Tabular type:

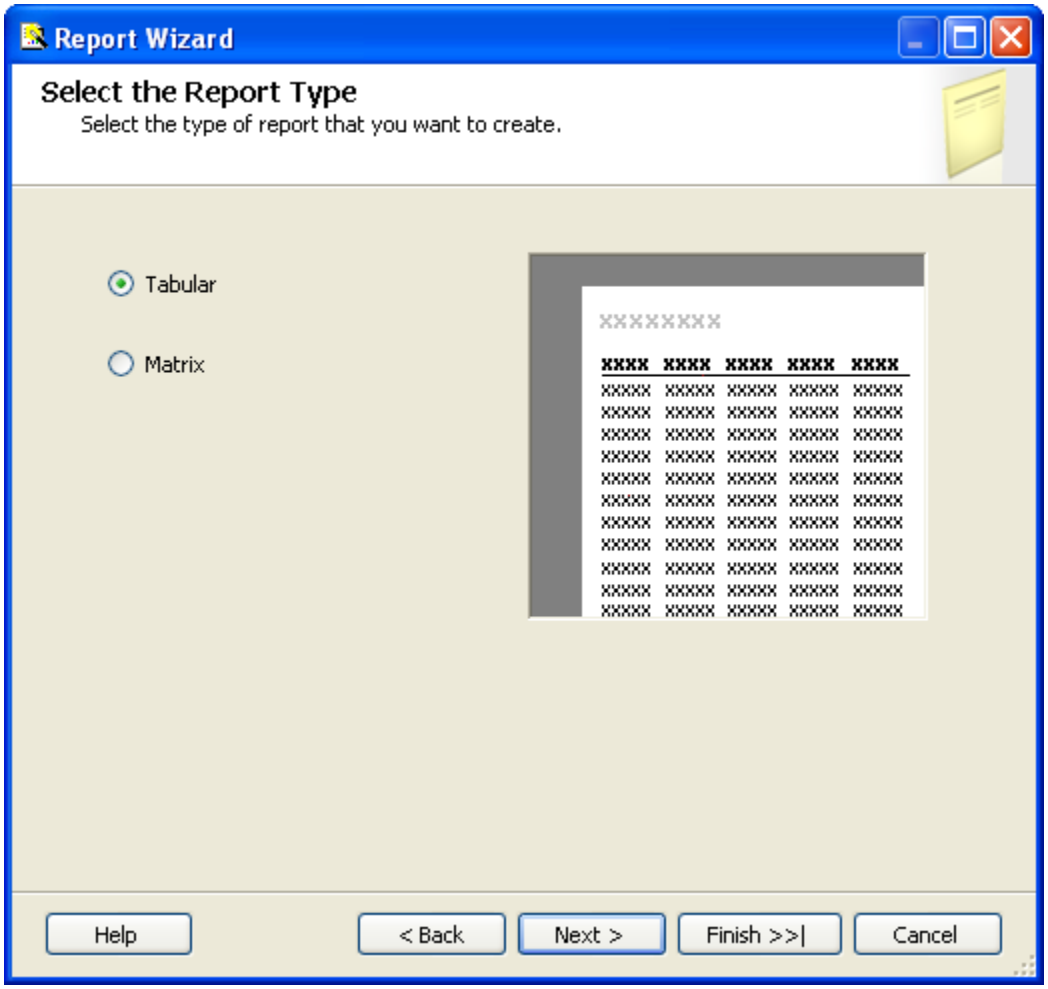


Figure 28.

©2014 Aras Corporation All Copyrights Reserved.

5.5 Design the Table

Select the fields that you want to show on the Report. In this case we were selective about which fields to return from the query select clause so we want all the fields.

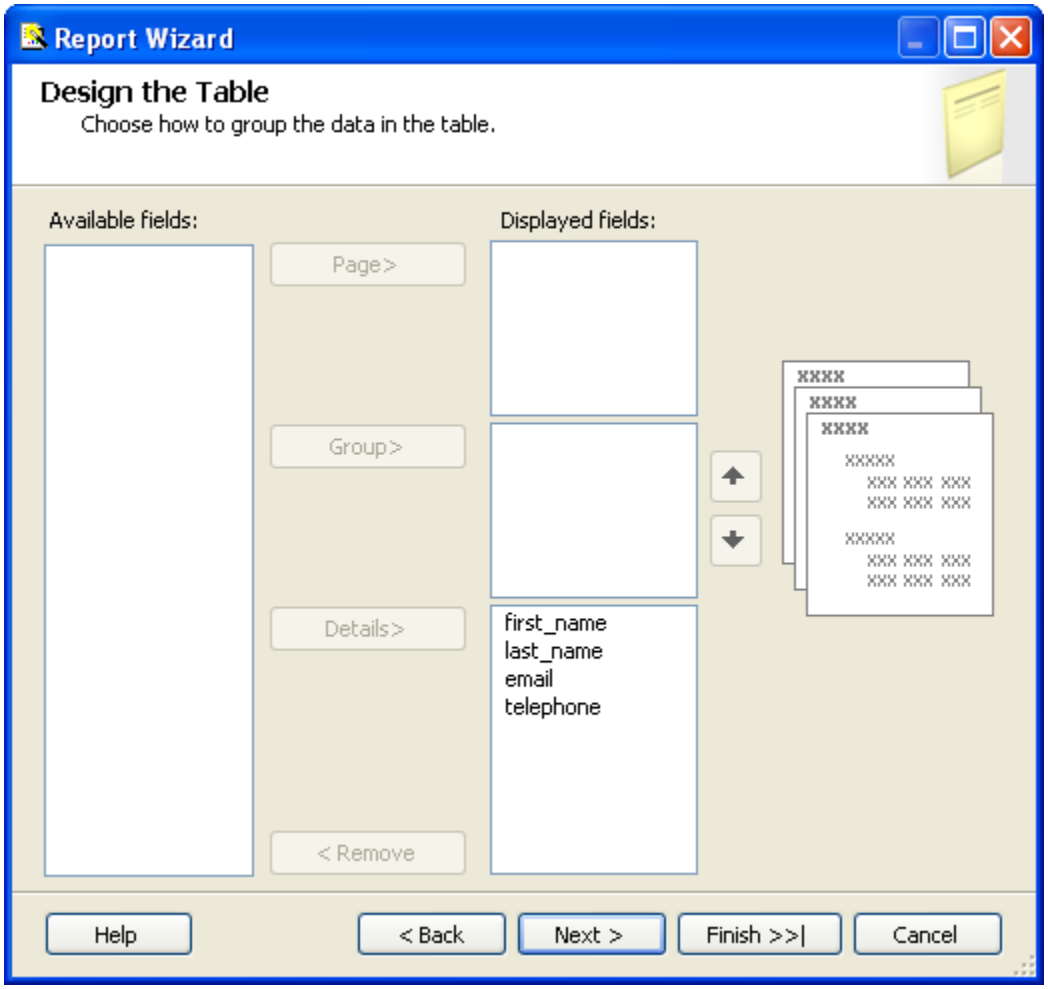


Figure 29.

©2014 Aras Corporation All Copyrights Reserved.

5.6 Choose the Table Style

Select the style for the table for the Report. This example simply uses the default bold style.

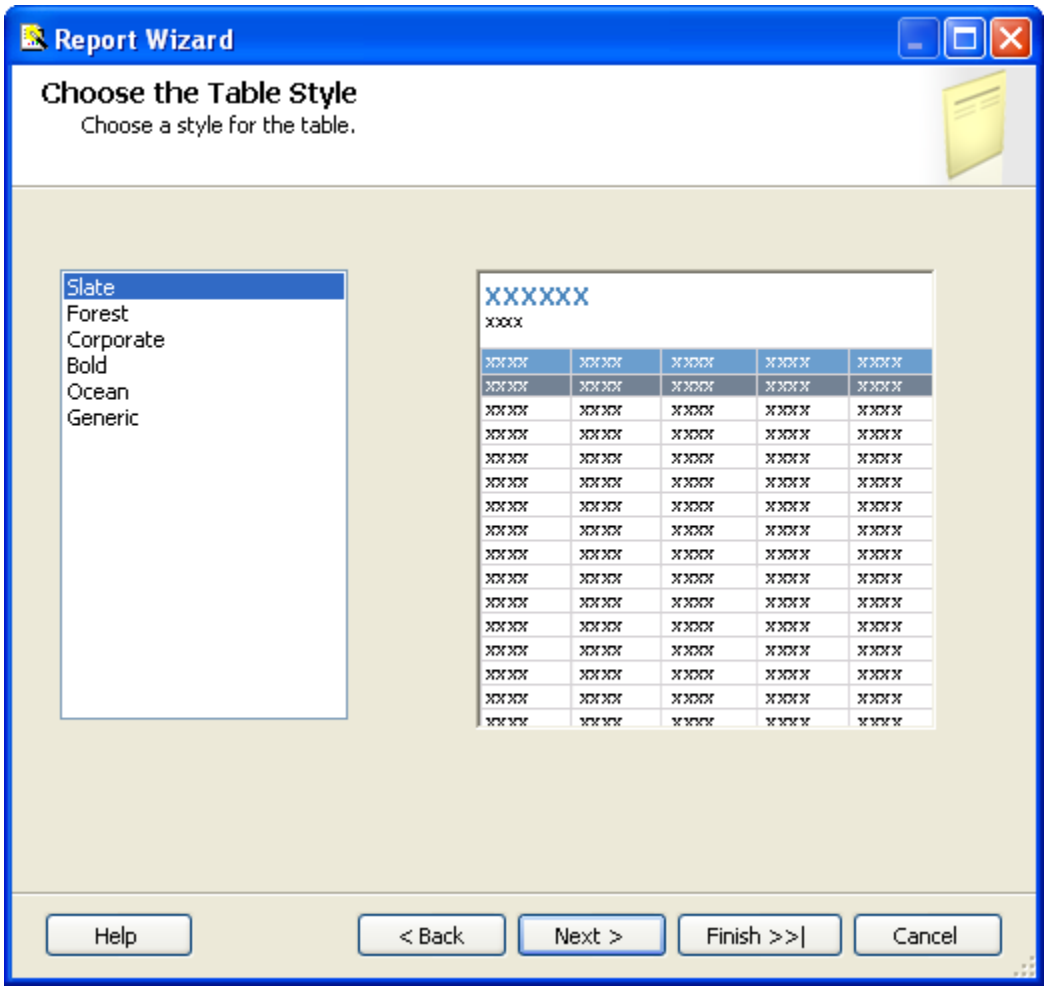


Figure 30.

©2014 Aras Corporation All Copyrights Reserved.

5.7 Name the Report

The final step in the Report wizard is to name the Report.

Note: The Report Name must be chosen to be meaningful and is the same name you set as the name of the Report Item in Aras Innovator.

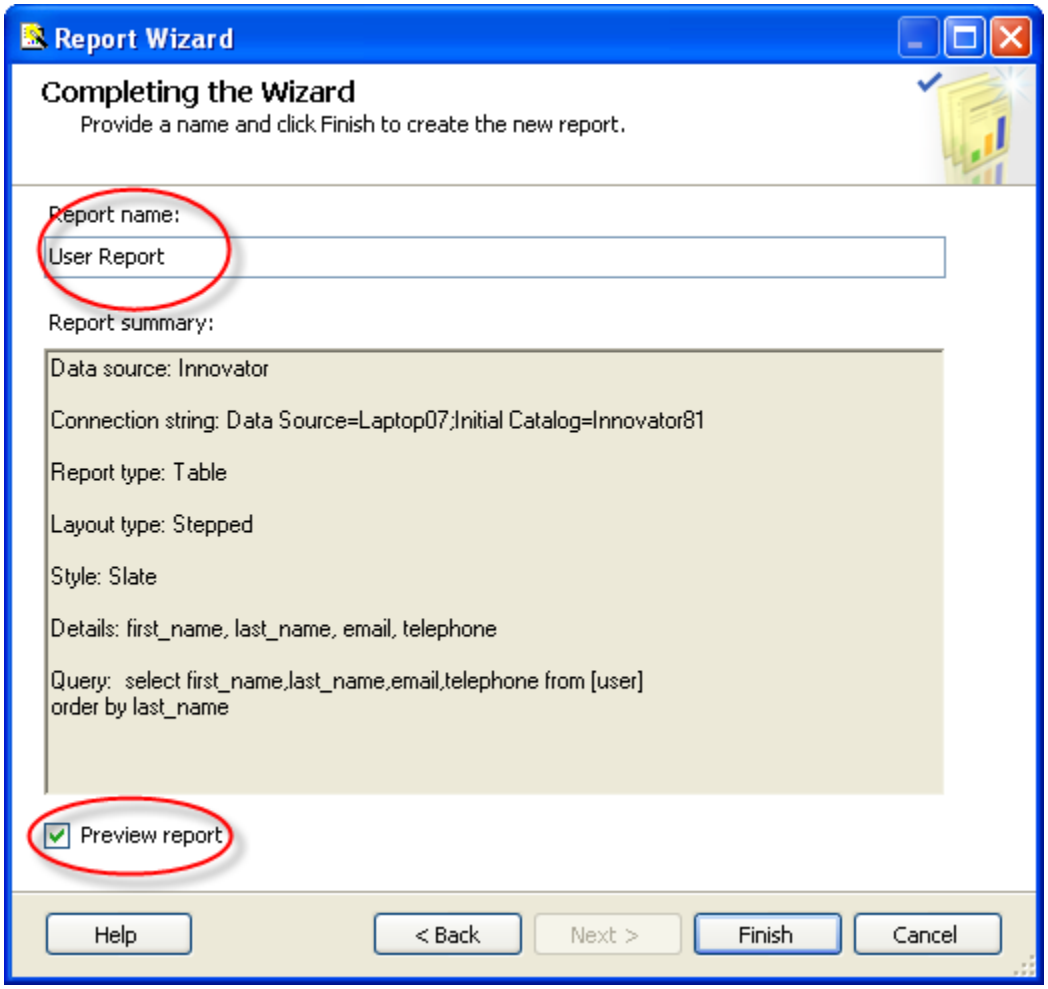
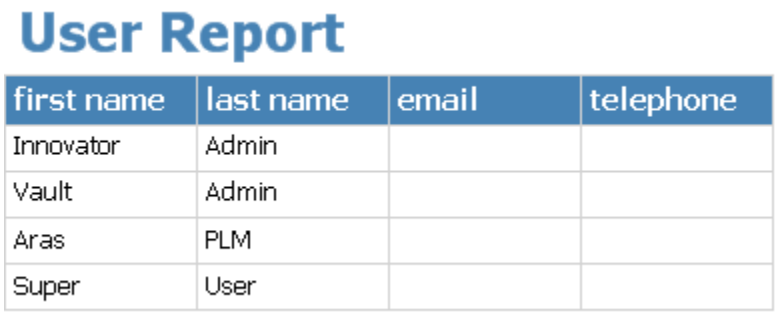


Figure 31.

©2014 Aras Corporation All Copyrights Reserved.

5.8 Wizard Report Results

The Report wizard builds the report, which look similar to the following screen shot:



User Report

first name	last name	email	telephone
Innovator	Admin		
Vault	Admin		
Aras	PLM		
Super	User		

Figure 32.

Once the Report is opened in the Layout Tab you can make further changes as required. In this example, we camel case the column headers. For example, first name becomes First Name and we resize the width of the table and the columns to take up 6.5 inches horizontally.

5.9 Visual Studio Layout

The Report pane has three Tabs:

- **Data**: For editing the query for the Report. Look at the appendix in this document to see a more complex Report query example, using the graphical query builder instead of the wizard.
- **Layout**: For visually designing the Report with drag and drop functionality.
- **Preview**: For viewing the finished report with real data.

The **Properties** panel on the right is used for setting fonts, color, and other visual properties.

The **Fields** panel on the left is used to drag database fields onto the report.

The **Toolbox** panel also on the left is for dragging Report controls onto the Report layout. This is how you can add tables, labels, images, and charts to the Report layout.

Note: Use the **View --> Fields** menu pick to open the Fields panel when in Layout mode.
Use the **View --> Toolbox** menu pick to open the Toolbox panel.
To open the Header and Footer, right-click to open the context menu in the pink field in the Report layout.

©2014 Aras Corporation All Copyrights Reserved.

5.10 Deploying the Report

Once you are satisfied with the Report query and layout as viewed from the Preview Tab, you need to deploy the Report to the Reporting Service. From the 'Solution Explorer' panel right-click on the 'User Directory' Report item and select the context menu choice Deploy. Watch for error messages in the output window at the bottom of the frame.

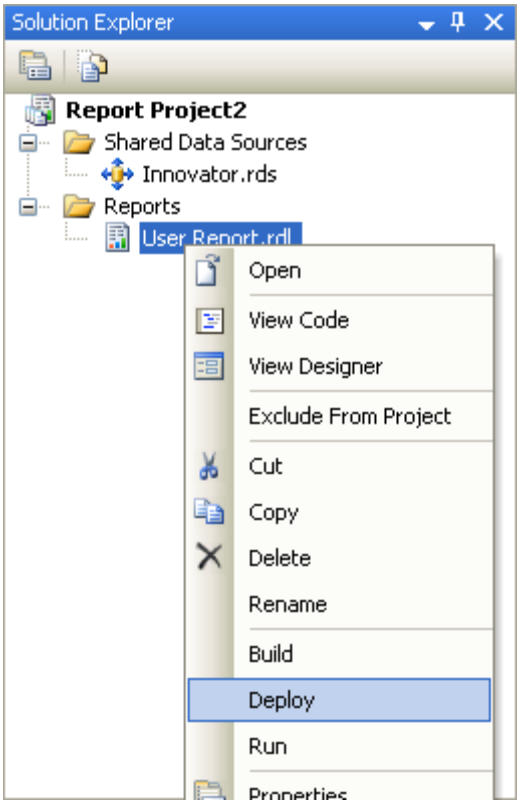


Figure 33.

5.11 Confirm Deployment

To confirm the new report has been deployed, open IE Browser and navigate to http://localhost:8080/Reports. You should see the 'Innovator Reports' folder. Click on that link and you should see the report you created

Select the hyperlink for the new Report and it is opened and run the Report.

Note: The URL for directly testing this report outside of Aras Innovator would look like this: http://localhost:8080/ReportServer?/Innovator Reports/User Directory.

5.12 Running the Report from within Aras Innovator

In order for users to run the new Reporting Services Report from within Aras Innovator you need to create an Innovator Report Item. The important settings to enable the Innovator/Reporting Services interface are:

- The Innovator Report Item name must be exactly the same as the Reporting Services Report name.

©2014 Aras Corporation All Copyrights Reserved.

- The Location Field choice must be set to the new '**Service**' option.
- The Report Query Field specifies the parameters that are passed to the Report Server.

5.12.1 Adding Aras Innovator Report Item

1. Log into Aras Innovator as admin.
2. In the TOC, navigate to **Administration --> Reports**.
3. Click **New** on the main Toolbar.
4. When the new Report Form opens, enter the name of the Report (again this is the name you specified when you created the Report in Visual Studio).
5. Select the Report Type, which in this case is '**Generic**' meaning this Report, is available always.
6. Select the Report Location, which in this case is '**Service**'.
7. Select the Report Target, which in the case we want '**Window**'.

   Because this sample Report is of type ItemType there is no Report query Value.

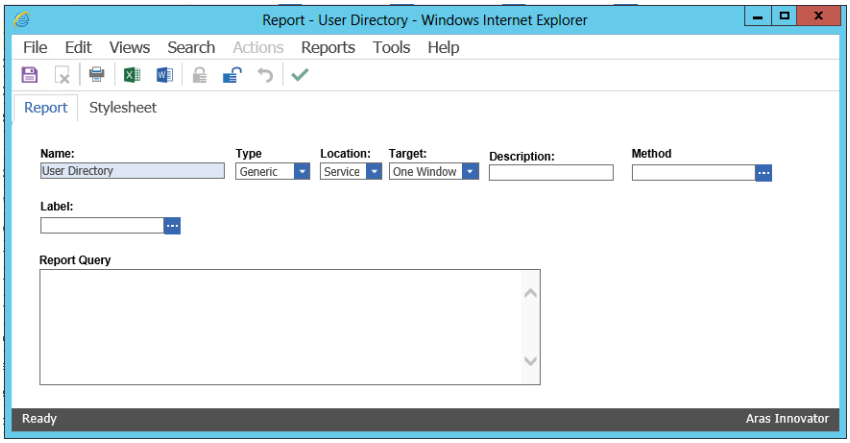


Figure 34.

Note: If you are using Firefox, you have to manually enter a stylesheet under the **Stylesheet** tab.

8. Test the Report by clicking on the **Reports** main menu bar choice and you should see the new '**User Directory**' choice. Click on the menu choice and you should see a new window open with the '**User Directory**' Report.

5.12.2 Passing Named Parameters to the ReportServer

As step 4 describes in the section 3.1 Report Server Variable, the value of the Innovator Report Item Property report_query is appended to the ReportServer URL providing additional parameters to control the ReportServer.

For example, the ID of the Item can be passed as the Named Parameter @ID for the SQL Select for the Reporting Services Report as in this example:

```
SELECT * FROM [PROJECT] WHERE id=@ID
```

To pass the ID for the context Item (the selected Item) on the URL to the ReportServer the Innovator Report Item report_query Property would look like this:

```
np:id=<xsl:value-of select="@id"/>
```

©2014 Aras Corporation All Copyrights Reserved.

This is actually an XSLT stylesheet fragment and is the directive for Innovator to add the string &id=ACBDEF0123456789ACBDEF0123456789 to the end of the ReportServer URL (but the actual ID value is used).

In this case the Innovator Report is of type Generic so no ID is required but perhaps we want the Reporting Services Report to be returned in PDF or Excel format. We can control the ReportServer by adding the following parameter as the report_query value for the 'User Directory' Report: rs:Format=PDF

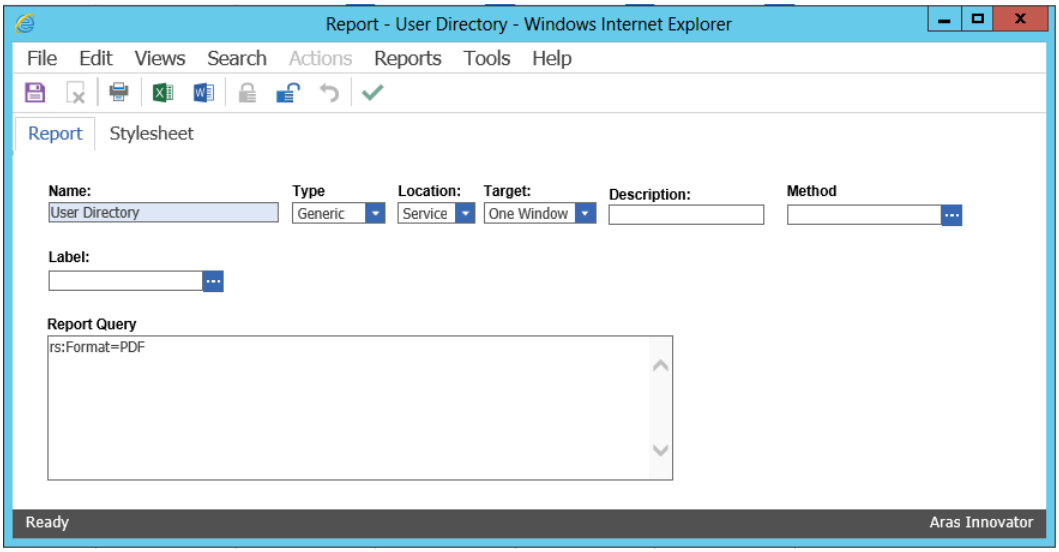


Figure 35.

Note: If you need to include more than one parameter for the report_query value, you need to use the Entity & rather than the actual & symbol to delimit the parameters. Recall that the report_query value is eventually part of the query_string for the URL to the ReportServer but is actually an XSLT stylesheet fragment and so the special symbols like <, >, and & need to be protected in XML need to be entered as Entities < > and & respectively.

For example, to pass the Named Parameter ID and also the Format PDF in the same report_query the value would look like this:

```
np:id=<xsl:value-of select="@id"/>&amp;rs:Format=PDF
```

©2014 Aras Corporation All Copyrights Reserved.

6 Report Server query_string Parameters

There are several parameters that you can use to control how the Reporting Services run your Report requests. You include these in the Innovator Report Item's report_query property, which the client then uses to form the query_string for the ReportServer request:

6.1 Named Parameters

Named Parameters passed onto the SQL Select query. For example,

```
np:last_name=Kohl
```

Would map the value Kohl to the Named Parameter @last_name in the SQL Select:

```
SELECT * FROM [USER] WHERE last_name=@last_name
```

6.2 Format Parameter

Format parameter to control the format the Report is rendered:

```
rs:Format=XML
rs:Format=PDF
rs:Format=HTML4.0
rs:Format=MHTML
rs:Format=EXCEL
rs:Format=Word
```

6.3 Report Toolbar

Turn off the Report Toolbar:

```
rc:Toolbar=false
```

6.4 Parameters Form

Turn off the Parameter Collection area above the Report, which is a simple dynamically created Form with Fields to allow the user to enter input for the Named Parameters for the SQL Select query. The Named Parameters passed on the ReportServer URL query_string are shown and the user fills in the rest:

```
rc:Parameters=false
rc:Parameters=Collapsed
```

6.5 Zoom

Control the zoom:

```
rc:Zoom=100
rc:Zoom=% zoom as integer
rc:Zoom=Page Width
```

©2014 Aras Corporation All Copyrights Reserved.

```
rc:Zoom=Whole Page
```

6.6 Reports Folder

The default Folder where Reporting Services Reports are deployed is in ‘Innovator Reports’ however, you may deploy the Reports in other Folders and you specify this using the this Paramater:

```
irs:Folder=Your Folder Name
```

6.7 Report Name

The Aras Innovator client automatically appends the Parameter to specify the Report to run on the query_string, which is the name of the Innovator Report Item. However, if run reports programmatically you may need to set this directly yourself:

```
irs:Report=Your Report Name
```

©2014 Aras Corporation All Copyrights Reserved.

7 Report Manager

The Reporting Services ReportServer comes with a Report Manager page to administer the Reports, Folders, Data Sources, and so on. This page has been integrated into Aras Innovator client via an Innovator Report that opens the Report Manager page (see *section 2 Configuring Aras Innovator* for more information about this Report Item).

Reports that are created and tested on the development machine can be moved to a production ReportServer by exporting the RDL file for the Report from your local development machine and then importing it into the production machine.

The process of exporting a Report and importing it into another ReportServer and enabling the Report in Aras Innovator is a four step process:

1. Export the RDL file via the Report Manager page.
2. Import the RDL file via the Report Manager page.
3. Reassign the Data Source to the imported Report.
4. Add an Innovator Report Item to call the Reporting Services Report.

7.1 Opening the Report Manager page

1. In the TOC, navigate to **Administration --> Reports**.
2. On the main menu, click **Reports** and then, **Report Manager** choice.
3. The Report Manager page opens in a new window as shown below:

   You may need to click **Show Details**.

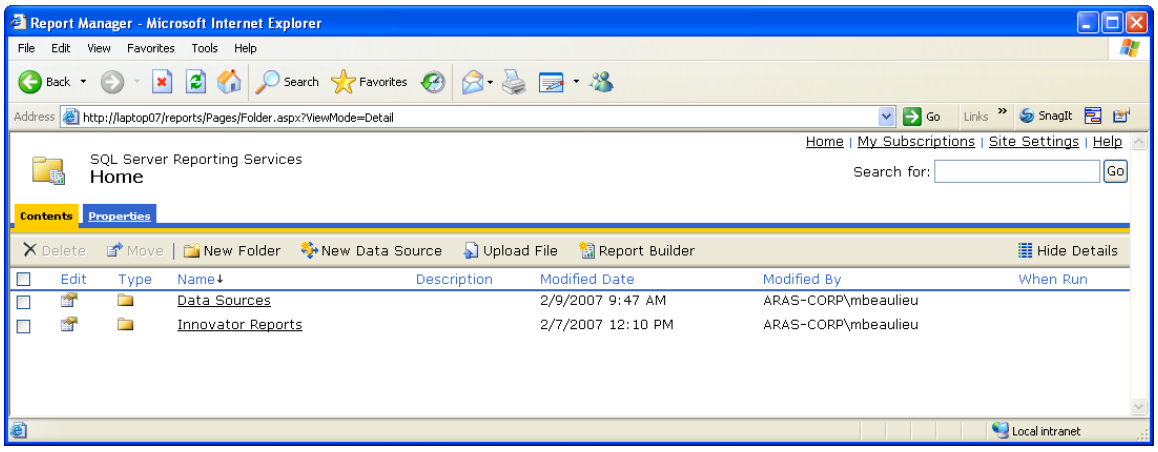


Figure 36.

Note: This same page can be opened from your browser with the URL http://localhost:8080/Reports.

©2014 Aras Corporation All Copyrights Reserved.

7.2 Export the RDL

1. To export a RDL file for a Report, click on the '**Innovator Reports**' Folder. You see a list of Reports and Shared Data Sources.

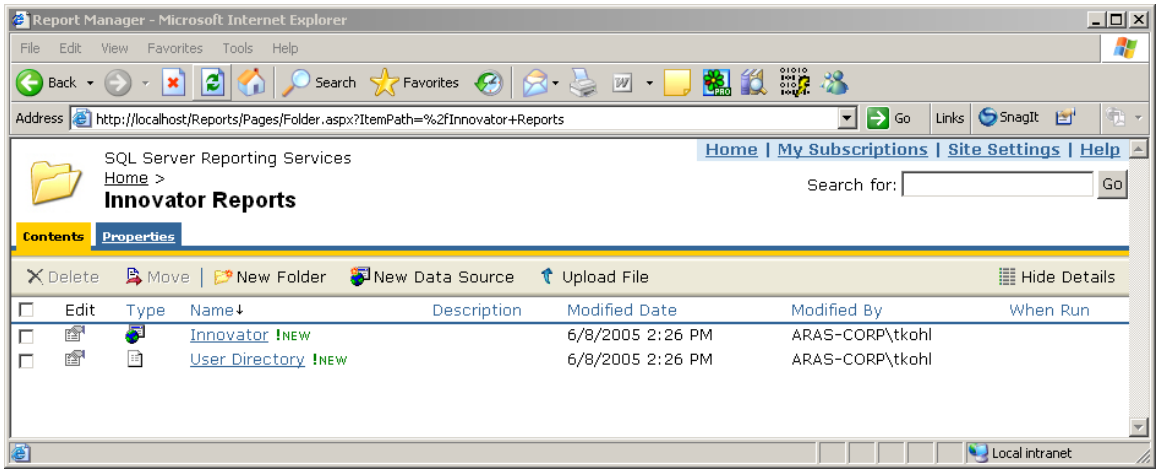


Figure 37.

2. Click on the properties icon in the Edit column for the Report you want to export. This opens the Properties Page for the Report.

3. Click on the '**Edit**' link below the 'Report Definition' label. This opens a dialog asking for confirmation that you want to save the RDL file.

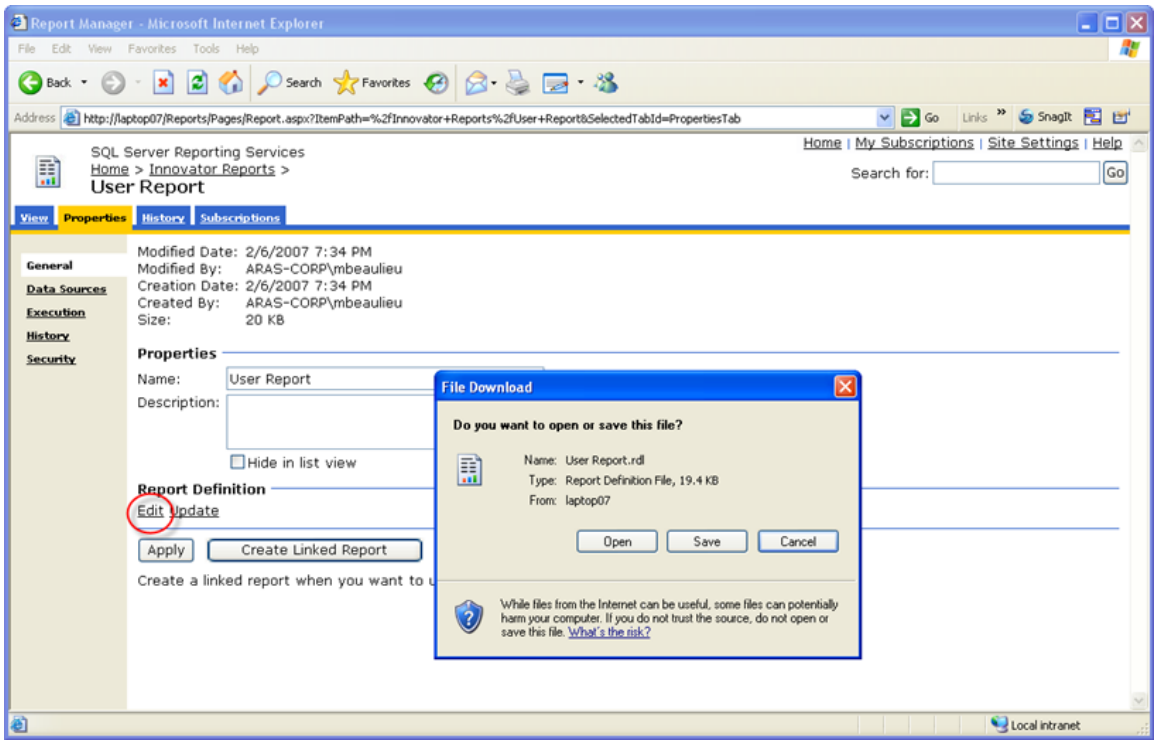


Figure 38.

©2014 Aras Corporation All Copyrights Reserved.

4. Click '**Save**', which opens a File '**Save As**' dialog.

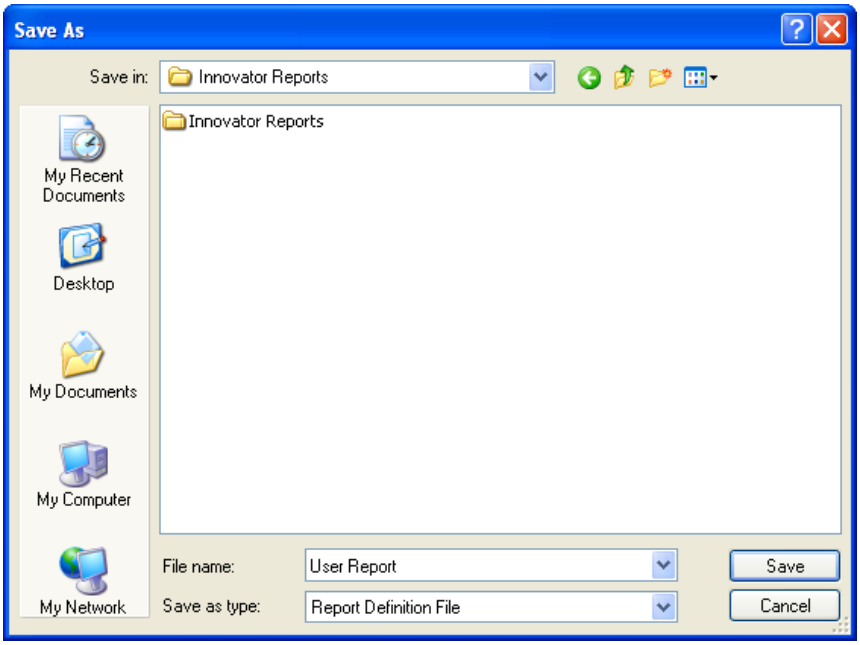


Figure 39.

7.3 Import the RDL

Log into the target Aras Innovator Server:

1. In the TOC, navigate to **Administration --> Reports**.
2. Click **Reports** on the main menu and the '**Report Manager**' choice.
3. The **Report Manager** page opens in a new window as shown below:

> **Note:** This same page can be opened from your browser with the URL http://localhost:8080/Reports.

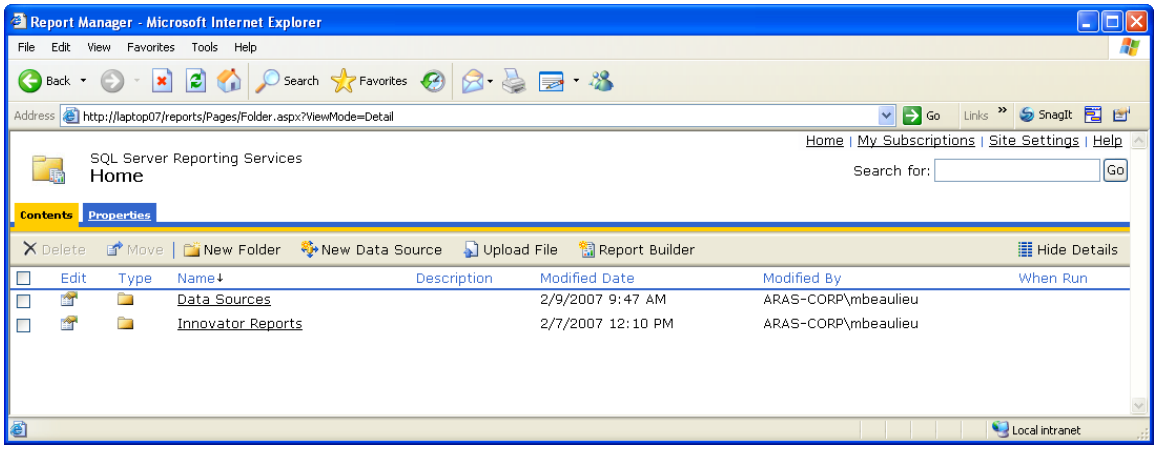


Figure 40.

4. Click on the '**Innovator Reports**' Folder and you see a list of Reports and Shared Data Sources. Click '**Upload File**' in the middle of the screen.

©2014 Aras Corporation All Copyrights Reserved.

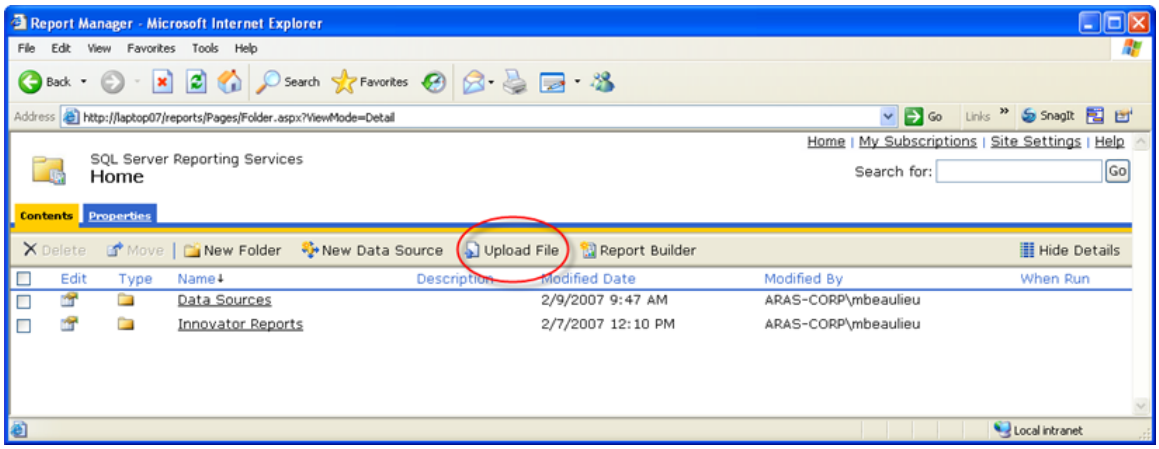


Figure 41.

This opens the Upload File page.

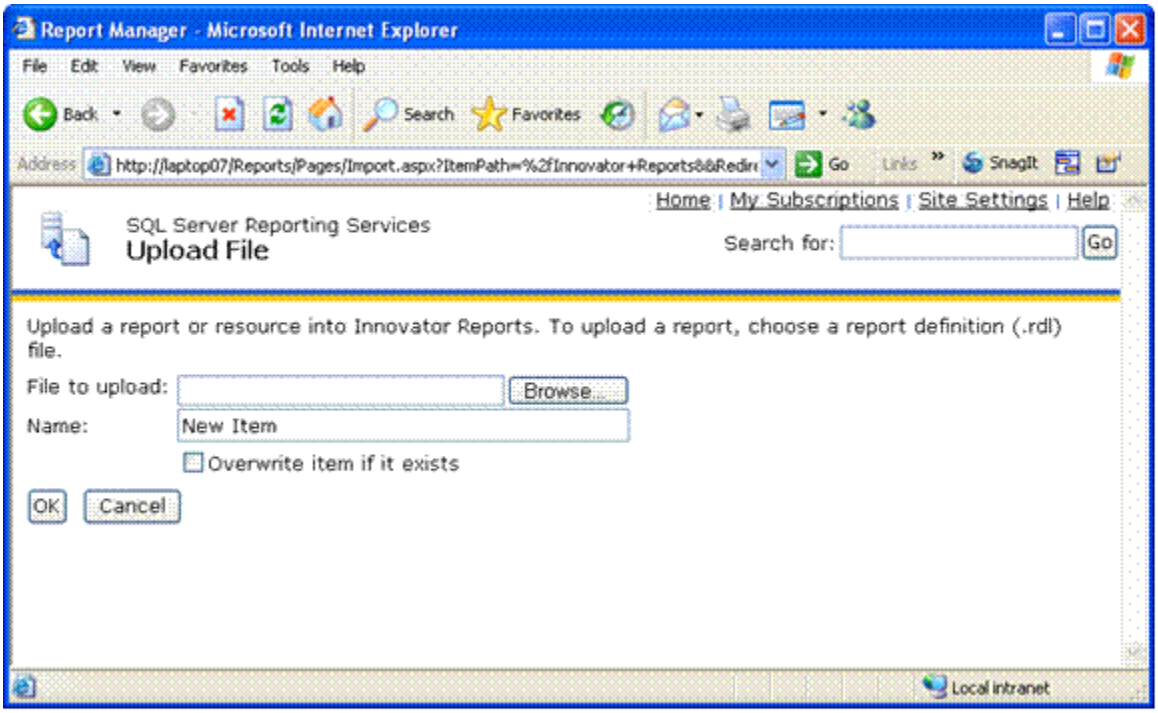


Figure 42.

5. Click '**Browse**…', which opens a File Selection dialog for you to select the RDL file to upload.
6. Select the RDL file and click '**Open**'. This inserts the file name in the File to upload field and also set the Name field to the name of the RDL file selected.

   If this is a new version of an existing Report select the Overwrite checkbox. Click '**OK**' to complete the upload. The RDL file is uploaded to the target ReportServer and deployed. You should see the new Report in the Folder now.

©2014 Aras Corporation All Copyrights Reserved.

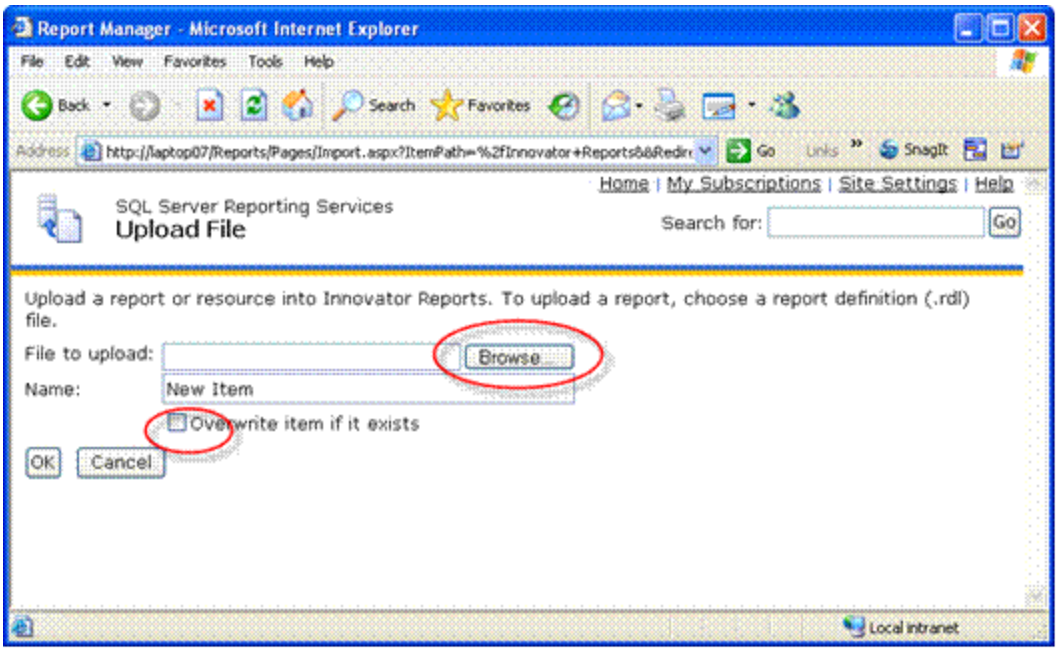


Figure 43.

7.4 Select the Shared Data Source

The Data Source for the Report may need to be changed if the name of the Shared Data Source is different on the target ReportServer. Click on the properties icon in the Edit column for the new Report that was uploaded. This opens the Properties page for the Report where you can select its Shared Data Source.

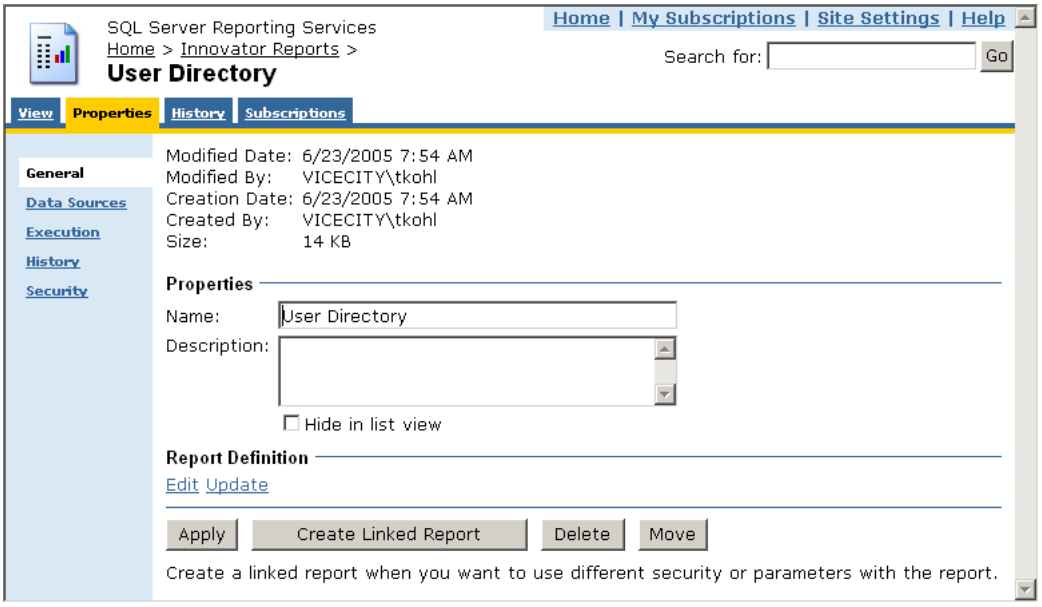


Figure 44.

Select the '**Data Sources**' navigation link. The first choice '**A shared data source**' should be selected.

©2014 Aras Corporation All Copyrights Reserved.

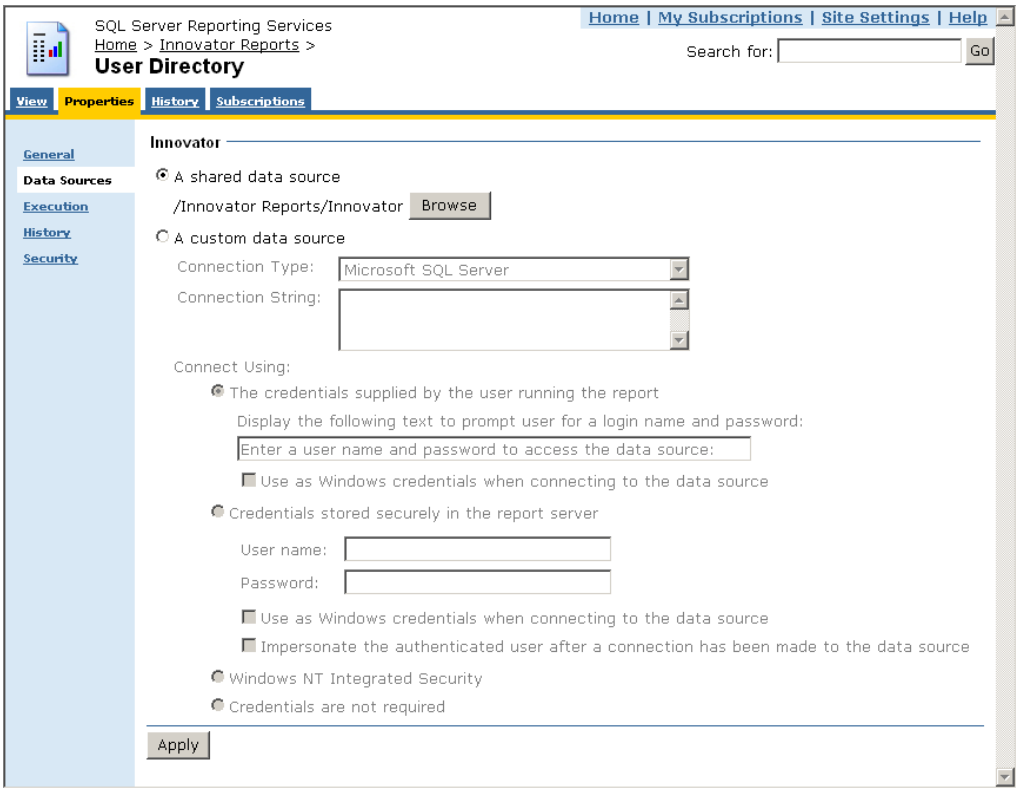


Figure 45.

If the named Shared Data Source is incorrect for this ReportServer click '**Browse**' to open the Shared Data Source page. Navigate the Folders to locate the desired Shared Data Source, select it, and click '**OK**'.

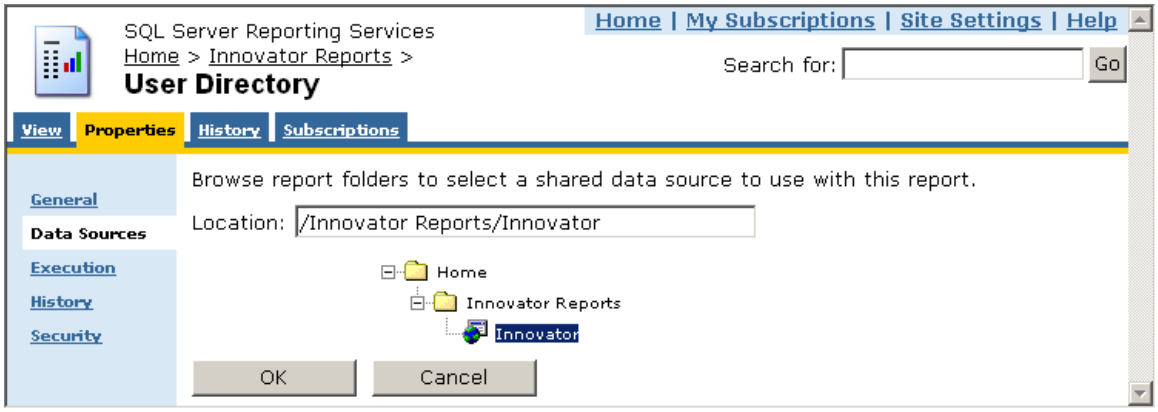


Figure 46.

©2014 Aras Corporation All Copyrights Reserved.

8 Graphical Query Builder

The Visual Studio report designer has a graphical query builder that can be useful when constructing reports for deep item structures such as the Project Item structure.

On the Data Tab in the report designer select the icon which opens the graphic design window.

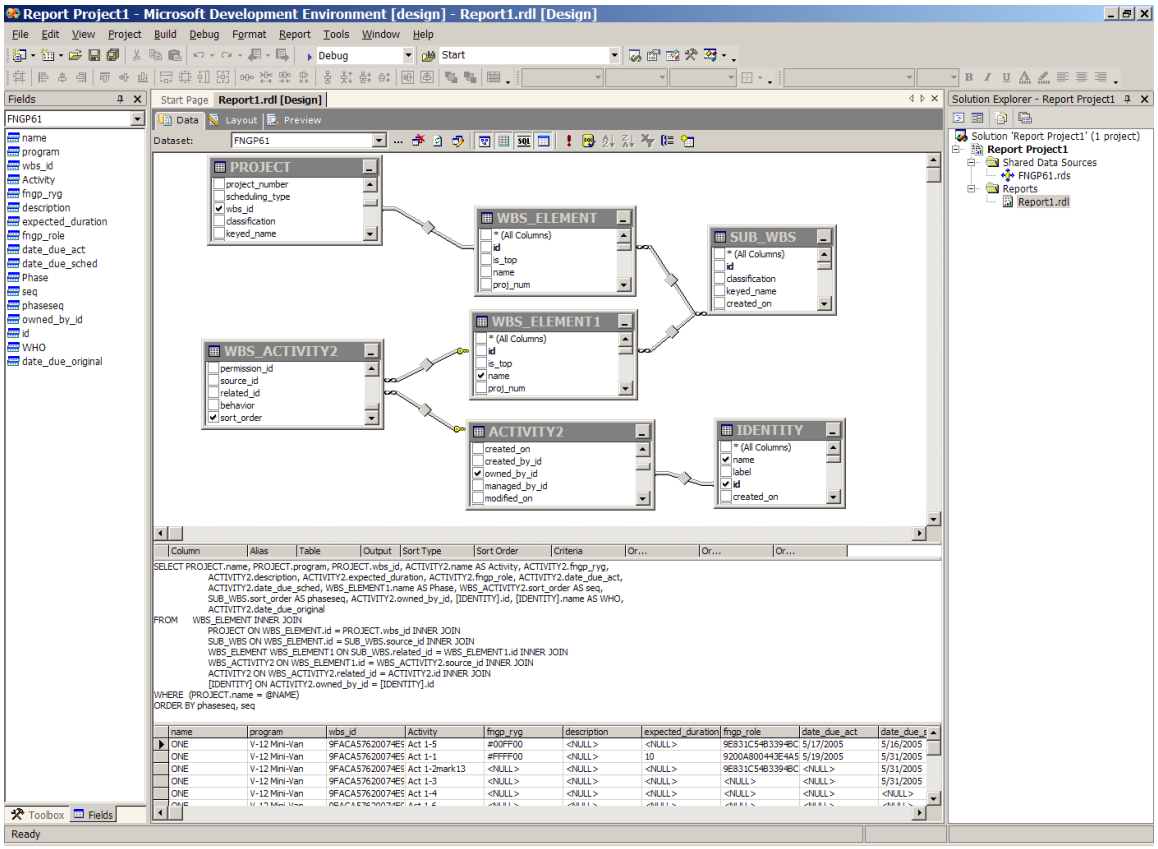


Figure 47.

Manipulating the graphics automatically creates the correct query with the joins.

©2014 Aras Corporation All Copyrights Reserved.

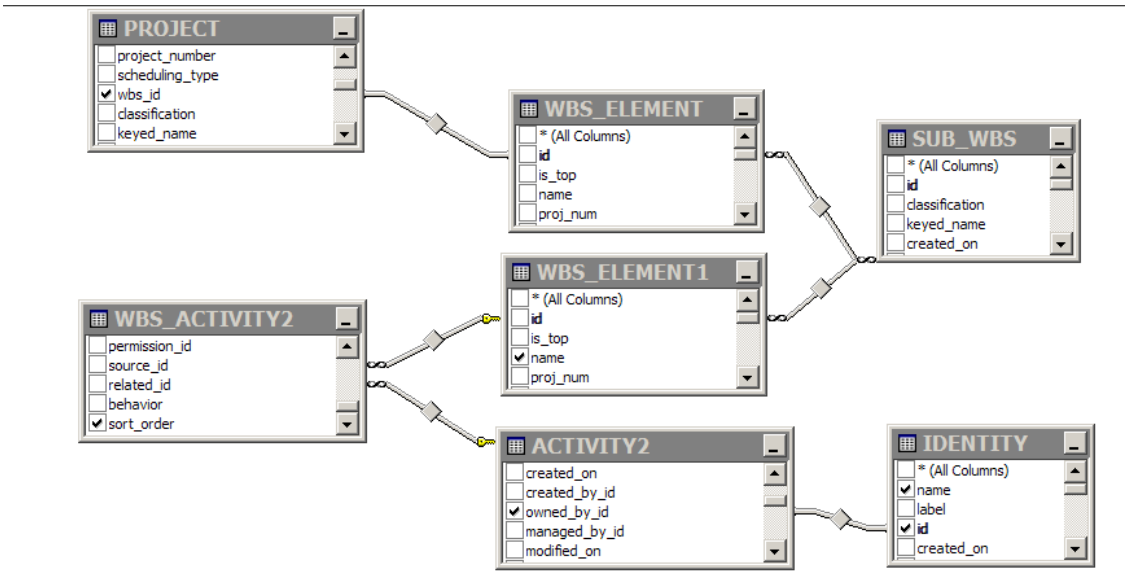


Figure 48.

This results in the SQL query:

```
SELECT PROJECT.name, PROJECT.program, PROJECT.wbs_id, ACTIVITY2.name AS
Activity, ACTIVITY2.fngp_ryg, ACTIVITY2.description,
ACTIVITY2.expected_duration, ACTIVITY2.fngp_role, ACTIVITY2.date_due_act,
ACTIVITY2.date_due_sched, WBS_ELEMENT1.name AS Phase,
WBS_ACTIVITY2.sort_order AS seq,  SUB_WBS.sort_order AS phaseseq,
ACTIVITY2.owned_by_id, [IDENTITY].id, [IDENTITY].name AS WHO,
ACTIVITY2.date_due_original
FROM      WBS_ELEMENT INNER JOIN
      PROJECT ON WBS_ELEMENT.id = PROJECT.wbs_id INNER JOIN
      SUB_WBS ON WBS_ELEMENT.id = SUB_WBS.source_id INNER JOIN
      WBS_ELEMENT WBS_ELEMENT1 ON SUB_WBS.related_id = WBS_ELEMENT1.id INNER
JOIN
      WBS_ACTIVITY2 ON WBS_ELEMENT1.id = WBS_ACTIVITY2.source_id INNER JOIN
      ACTIVITY2 ON WBS_ACTIVITY2.related_id = ACTIVITY2.id INNER JOIN
      [IDENTITY] ON ACTIVITY2.owned_by_id = [IDENTITY].id
WHERE  (PROJECT.name = @NAME)
ORDER BY phaseseq, seq
```

©2014 Aras Corporation All Copyrights Reserved.

9 Date Related SQL Queries

AML is the language of Aras Innovator, but some external add-ons do connect to Aras Innovator on a direct SQL level. Because Aras Innovator stores DateTimes in SQL Server as UTC, we have provided an add-on function for Microsoft SQL Server that allows the easy conversion of DateTime to and from UTC.

To convert DateTime values from UTC to a specific time zone, use the function ConvertToLocal ({value},{Time Zone}). The time zone should be specified according to the registry key name of the time zone desired. (See *Aras Innovator - Configuring Internationalization*) To use the DEFAULT parameter for the {Time Zone}, it is required that a value be set for CorporateTimeZone.

Example:

```
select item_number, created_on

from innovator.Document
```

Would be written as

```
select item_number, innovator.ConvertToLocal(created_on,DEFAULT) as
CreadtedOn

from innovator.Document
```

Or

```
select item_number, innovator.ConvertToLocal(created_on,'Eastern Standard
Time') as CreadtedOn

from innovator.Document
```

To convert DateTime values from a specific time zone to UTC, use the function ConvertFromLocal({value},{Time Zone}). The time zone should be specified according to the registry key name of the time zone desired. (See *Aras Innovator - Configuring Internationalization*) To use the DEFAULT parameter for {Time Zone}, it is required that a value be set for CorporateTimeZone.

Example:

```
update innovator.Documnet

set effective_date = '1/1/2007 00:00:00.00'
```

Would be written as

```
update innovator.Documnet

set effective_date = innovator.ConvertFromLocal('1/1/2007
00:00:00.00',DEFAULT)
```

Or

```
update innovator.Documnet

set effective_date = innovator.ConvertFromLocal('1/1/2007
00:00:00.00','Eastern Standard Time')
```

This allows users to write queries in a familiar time zone context.

```
select item_number, innovator.ConvertToLocal(release_date,DEFAULT) as
ReleaseDate

from innovator.Document

where created_on >= innovator.ConvertFromLocal('1/1/2007
00:00:00.00',DEFAULT)
```

©2014 Aras Corporation All Copyrights Reserved.

10 Cookbook

This section provides a set of recipes for doing common reporting tasks.

10.1 User Input Reports

You need a Form to collect user input for Named Parameters for the SQL query for Reporting Services based Reports.

Technique

An XSLT based Report is used to present a Form to collect the user input. The user input is mapped to a query_string for the Report Server request via the RSGatwway.aspx page.

In the ReportTool/Templates folder there is a UserInputTemplate.txt file that you can use as a template to build the XSLT based Form. This provides a generic way to define the Fields for the user input Form, which opens a modal dialog with a generic HTML page dynamically building the Form based on the Fields defined in the template file.

Note: To use 'User Input Reports' you must add the following script to the UserInputDialog.html located in the ReportTool folde and save itr:

```
<script type="text/javascript"
src="../../javascript/AddConfigurationLink.js"></script>
W
```

Procedure for building a user input Form for Reporting Services based Reports with Named Parameters:

1. Login to Aras Innovator as admin.
2. In the TOC, navigate to **Administration --> Reports**.
3. Open a new Report Item.
4. Set the Report Item properties as following:
   a. **Name** = the name of your Report as deployed in Visual Studio to the Report Server.
   b. **Type** = Generic or ItemType depending on when you want the Report to appear on the Report menu; Generic always, ItemType only when that ItemType is selected off the Main Tree.
   c. **Location** = Server; this is required because the XSLT page returns an HTML page with JavaScript that needs to be parsed and Client side Report only writes the transformed page to the browser window and thus the JavaScript is not parsed.
   d. **Target** = Window; again because the page has JavaScript we need a window to parse the JavaScript. This page presents status info to the user as the Report is being processed and the final Report results are shown in this same window.
   e. **Report Query** = leave blank it is currently not used for this purpose.
5. Click on the Stylesheet Tab to insert the template code for setting up the Fields for the modal dialog:
   a. Scroll to the bottom of the page.
   b. Open the ReportTool/Templates/UserInputTemplate.txt file and select the whole document content.
   c. Replace the <body> tag in the Stylesheet Tab with the content from the UserInputTemplate.txt file.

©2014 Aras Corporation All Copyrights Reserved.

d. Edit the userInputFields object, which is an IOM Item by setting the properties for the Item to match the Named Parameters for the SQL query for the Report as deployed to the Report Server.

```
///
/// Set the User Input Fields...
/// The property names must match exactly the names you used for the
/// Named Parameters in the SQL Query.
///
/// For example, this SQL Query for the GetParts Report in the
/// Report Server has the following
/// Named Parameters: part_number, description, and cost;
///
/// SELECT     part_number, description, cost
/// FROM         MY_PART
/// WHERE     (part_number LIKE @part_number)
///       AND    (description LIKE @description)
///       AND    (cost LIKE @cost)
///
var userInputFields = opener.top.myInnovator.newItem();
userInputFields.setProperty('part_number', '%');
userInputFields.setProperty('description', '%');
userInputFields.setProperty('cost', '%');
```

e. Click **Apply** at the bottom of the Stylesheet Tab.
f. Click **Save** on the Report Tool toolbar.

That's it, the Report Item should now appear on the Report menu and when selected the User Input Dialog should open. When the user clicks **Submit** the user input is mapped as a query_string and the actual Report is requested from the Report Server via the RSGateway.

If the future the Named Parameters and Report name are derived and the XSLT based Report Stylesheet is automatically generated but for now you need to set these values when using the template.

If you want to add validation or rearrange the Fields on the Form you can make a copy of the ReportTool/UserInputDialog.html page and modify it to meet your needs.

If you have the need to pass attributes or property values from the selected context Item use the following technique to set the values to hidden form fields:

1. Edit the section from the UserInputTemplate.txt file to pack the desired Item attributes and properties into hidden form fields:

```
<!--
  The following is an example for passing attributes and properties
  from the selected context Item as hidden fields.

  For example, pack the id attribute from the context Item using
  XSLT Attribute Substitution and nested xsl tags for the properties.

  <form>
    <input type="hidden" name="id" value="{//Item/@id}/>
    <input type="hidden" name="name">
      <xsl:attribute name="value" select="//Item/name"/>
```

©2014 Aras Corporation All Copyrights Reserved.

```
    </input>
  </form>
  -->
```

2. Also edit the section from the UserInputTemplate.txt file to pass those hidden field values as Field values for the user input dialog:

```
///
/// If there are properties from the selected context Item packed
/// as hidden fields on the form use this:
///
userInputFields.setProperty('id', document.form[0]['id'].value);
userInputFields.setProperty('name', document.form[0]['name'].value);
```

10.2 Report Chooser

You need a dialog to present a set of Reporting Services based Reports to choose from for the selected Item.

Technique

An XSLT based Report is used to present a dialog to get the Report choice. The user selection is mapped to a query_string for the Report Server request via the RSGatwway.aspx page. This is similar to recipe 10.1 User Input Form but in this case we do want the ID for the selected Item to be passed as a Named Parameter on the query_string.

In the ReportTool/Templates folder there is a file ChooseReportTemplate.txt that you can use as a template to build the XSLT based Form. This template provides a generic way to define the Reports choices and it opens a modal dialog.

Procedure for building a Report Chooser dialog for Reporting Services based Reports:

1. Login to Aras Innovator as admin.
2. In the TOC, navigate to **Administration --> Reports**.
3. Open a new Report Item.
4. Set the Report Item properties as following:
   a. **Name** = the name of your Report as deployed in Visual Studio to the Report Server.
   b. **Type** = Item (You do not need the report_query template when prompted).
   c. **Location** = Server; this is required because the XSLT page returns an HTML page with JavaScript that needs to be parsed and Client side Report only writes the transformed page to the browser window and thus the JavaScript is not parsed.
   d. **Target** = Window; again because the page has JavaScript we need a window to parse the JavaScript. This page presents status info to the user as the Report is being processed and the final Report results are shown in this same window.
   e. **Report Query** = leave blank it is currently not used for this purpose.
5. Click on the Stylesheet Tab to insert the template code for setting up the Report choices for the modal dialog:
   a. Scroll to the bottom of the page.
   b. Open the ReportTool/Templates/ChooseReportTemplate.txt file and select the whole document content.
   c. Replace the <body> tag in the Stylesheet Tab with the content from the UserInputTemplate.txt file.
   d. Edit the AddReport() calls to the Report choices:

©2014 Aras Corporation All Copyrights Reserved.

```
///
/// Add your Report Choices here...
///
AddReport('Report1', 'Report One', 'rs:Format=Excell');
AddReport('Report2', 'Report Two', 'rs:Format=HTML4.0');
AddReport('Report3', 'Report Three', 'rs:Format=PDF');
```

e. Click **Apply** at the bottom of the Stylesheet Tab.

6. Click **Save** on the Report Tool toolbar.
7. In the TOC, navigate to **Administration --> Reports**.
8. Search for the ItemType for this Report.
9. Edit the ItemType and select the Reports Tab.
10. Add an Existing related Item and select this Report.
11. Click **Save** on the ItemType toolbar.

That's it the Report should now appear on the Report menu, when the Item is selected. When the user clicks **Submit**, the selected Report the actual Report is requested from the Report Server via the RSGateway.

The ID for the selected Item is automatically passed as a Named Parameter np:id so you must name your Named Parameter in the SQL query with lowercase id also.

10.3 Date Selector

You need a Form with two date fields to collect user input for Named Parameters for narrowing the results of a SQL query for Reporting Services based Reports.

Technique

A Client-Side Report is used to call JavaScript, which presents the user with a dialog for selecting a date range. Upon the selection of the date range, the user clicks '**Run Report'** which calls the RSGateway.aspx page directly passing the date range as named parameters to the report query.

Procedure for building a Date Selector dialog:

1. Log into Aras Innovator as an administrator.
2. Navigate to **Administration --> Methods** and create a new method
   a. Name = run_report_button_click
   b. Type = JavaScript
   c. Method Code =

```
var fromDate = document.forms[0].from_date.value;
var toDate = document.forms[0].to_date.value;
if(new Date(toDate) < new Date(fromDate))
{
  alert("To Date must be greater than from date");
}
else
{
  var retVal=new Array();
  retVal["runReport"]="1";
```

©2014 Aras Corporation All Copyrights Reserved.

```
  retVal["to_date"] = toDate;
  retVal["from_date"] = fromDate;
  window.returnValue = retVal;
  window.close();
}
```

3. Save, Unlock, and Close method
4. Navigate to **Administration --> Forms** and create a new Form
   a. Name=DateRangeSelector
5. Add a new field
   a. Name=form_setup
   b. Field Type = HTML
   c. Label = <empty string>
   d. HTML Code

```
<font size="6">Date Range Selector</font>
<script>
document.isEditMode=true;
var myInnovator = dialogArguments.innovator;
setTimeout('startup();',100);

function startup()
{
var displayDate =  new Date();
document.forms[0].to_date.value=fixDate(displayDate);
document.forms[0].from_date.value=fixDate(displayDate);
}
function fixDate(inDate)
{
   var Day = inDate.getDate();
   var DayStr = (Day < 10) ? '0' + Day : Day;
   var Month = inDate.getMonth();
   Month++;
   var MonthStr = (Month < 10) ? '0' + Month : Month;
                     var cntx = myInnovator.getI18NSessionContext();
                     return cntx.ConvertFromNeutral(
inDate.getFullYear() + "-" + MonthStr + "-" + DayStr, 'date',
'short_date');
}

</script>
```

6. Add a new field
   a. Name = from_date
   b. Type=Date
   c. Label = From Date
7. Add a new field
   a. Name = to_date

©2014 Aras Corporation All Copyrights Reserved.

b. Type = Date
c. Label = To Date

8. Add a new field

a. Name = run_report
b. Type = Button
c. Label = Run Report
d. Add a new Field Event

1. Method Name = run_report_button_click
2. Event = onClick

9. Save, Unlock and Close Form.

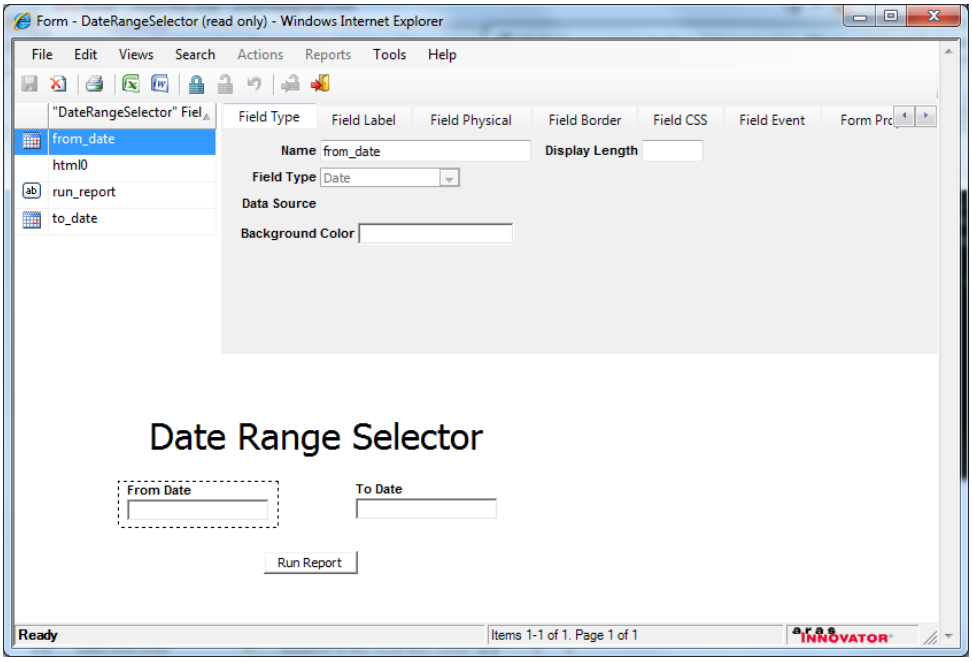


Figure 49.

10. Navigate to **Administration --> Methods** and create a new method

a. Name = Report_Date_Select
b. Method Code:

```
var param = new Object();
param.title  = "Date range selector for Reporting Services Report";
param.formId = "A0DC688E91834EC98E013A8046F60692"; //ID ofDate Range
Selector Form
params.isEditMode = "1";
param.aras   = top.aras;
param.innovator = this.getInnovator();
var res = showModalDialog("ShowFormAsADialog.html", param,
        "dialogHeight:300px; dialogWidth:450px; " +
        "status:0; help:0; resizable:0; scroll:0;");
// debugger;
if (res)
{
```

©2014 Aras Corporation All Copyrights Reserved.

```
var fromDate = top.aras.convertToNeutral(res["from_date"], 'date',
"short_date");
fromDate= fromDate.substring(0,fromDate.indexOf("T"));
var toDate = top.aras.convertToNeutral(res["to_date"], 'date',
"short_date");
toDate= toDate.substring(0,toDate.indexOf("T"));

var reportName = "TestReport";  //Your Report Name Here
var url   = top.aras.getServerBaseURL() +
"RSGateway.aspx?irs:Report=" + reportName;

//Date Variables in this example is fromDate and toDate
url += "&amp;np:fromDate=" + fromDate;
url += "&amp;np:toDate=" + toDate;
//url += "&amp;rs:Format=Excel";

var w = window.open(url, "TestReport",
"width=800,height=600,status=yes,resizable=yes,toolbar=no,menubar=no
,scrollbars=yes");
}
return "<html></html>";
```

11. Save, Unlock, and Close method.
12. Navigate to **Administration --> Reports** and create a new Report.
    a. SSRS Date Selector
    b. Type = Generic
    c. Location = Client
    d. Target = None
    e. Method = Report Date Select
13. Save, Unlock and Close report.

The new generic report should appear in Reports Menu. Your SQL query now accepts the name parameters of fromDate and toDate into the SQL query as a way to narrow your results in a where clause.

You must simply modify the Report_date_Select method to point to the proper Report and use the proper ID of the DateRangeSelector form.

©2014 Aras Corporation All Copyrights Reserved.