

Tracker Reports Documentation

Table of Contents

System Requirements.....	2
Getting Started	2
Report and Chart Solutions	2
Documentation	3
Templates	3
Model	5
Deploying Reports	5
Visual Studio/SQL Server Data Tools for Visual Studio Deployment.....	5
Data Source	5
Deployment Properties.....	7
Completing the Deployment	9
Model Deployment	10
Import Files from Report Manager	11
Logging into Report Manager.....	11
Uploading Individual Reports/Charts	12
Data Source	16
Model Deployment	18
Accessing Reports and Charts	19
Logging into Report Manager.....	20
Running Reports	21
Setting up Subscriptions	24
Creating Linked Reports	27
Ad-Hoc Reporting and Report Builder	32

This document describes the design and deployment of the *Tracker Reports* product. While it covers the basics on editing reports and deploying them to *SQL Server 2008 Reporting Services*, some knowledge of Reporting Services will be required, particularly for enterprise deployments. In addition, advanced users that want to customize or create new reports will likely need additional references. One that we have found to be excellent is *Microsoft® SQL Server™ Reporting Services* by Brian Larson (ISBN 978-0-07-154808-3).

System Requirements

Tracker Reports is a set of SQL Server 2008 Reporting Services reports, data sources, and models. As such, your database needs to be SQL Server 2008 or later. In addition, Reporting Services needs to be installed. If you want to test on your local machine, you can install the free SQL Server 2008 or 2012 Express. It has the ability to show reports but does not have subscriptions (emailing reports on a scheduled basis) or ad-hoc reporting. If you have an older version of SQL Server, you can view and create reports within *Visual Studio* or *SQL Server Data Tools for Visual Studio*¹, but you cannot actually deploy them.

Getting Started

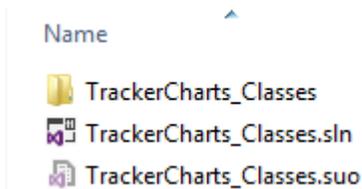
When you run the *Tracker Reports* installation, it will copy this document and all the report solutions to your local computer. This is shown to the right.

Report and Chart Solutions

As you might guess from the names, most of the directories correspond to either Reports or Charts for each of the main entities in Tracker.Net:

- Classes
- Courses
- Divisions
- Lessons
- Organizations
- Students

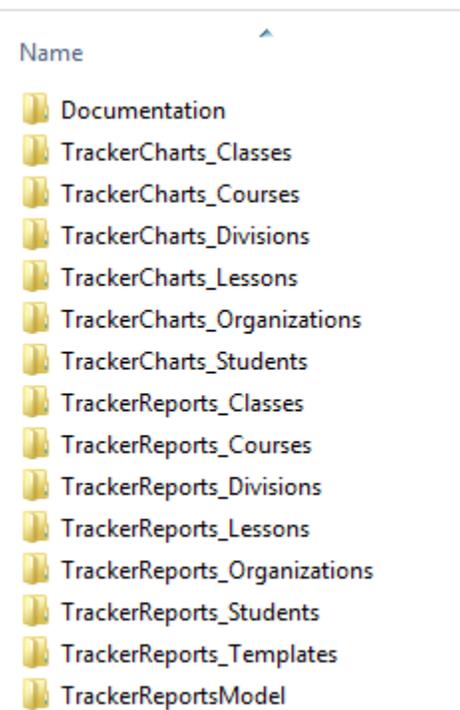
These directories contain a Solution file (.sln) as well as a subdirectory containing the files that go with it. This is shown to the left for the



the left for the

TrackerCharts_Classes

solution. The *TrackerCharts_Classes.sln* file is the one that you would open in *Visual Studio* or *SQL Server Data Tools for Visual Studio*. Within the *TrackerCharts_Classes* directory are the reports² themselves (*Courses Per Class.rdl* and the other *.rdl files), the Data



¹ We will cover these tools later in this document.

² Both charts and reports are *.rdl files. The distinction is the way that we have chosen to display the data. As you will see later, we recommend separating the charts and reports. But this is optional. You can combine them if desired.

Tracker Reports Documentation

Source (*TrackerDataSource.rds*), the report graphic (*trackernet.gif*), and various temporary data files (*Courses Per Class.rdl.data* and other *.data files). These latter files are only used for display when in development. The file list (minus the *.data files) is shown below.

Name	Type
 TrackerCharts_Classes.rptproj.user	Visual Studio Proj...
 TrackerCharts_Classes.rptproj	Report Project File
 TrackerDataSource.rds	RDS File
 Time Per Student.rdl	RDL File
 Time Per Student - Pie.rdl	RDL File
 Time Per Course and Class.rdl	RDL File
 Time Per Course and Class - Pie.rdl	RDL File
 Students Per Course and Class.rdl	RDL File
 Students Per Course and Class - Pie.rdl	RDL File
 Students Per Class.rdl	RDL File
 Students Per Class - Pie.rdl	RDL File
 Students Per Class - Breakout.rdl	RDL File
 Student Course Status.rdl	RDL File
 Student Course Status - Pie.rdl	RDL File
 Student Course Lesson Status.rdl	RDL File
 Student Course Lesson Status Consolidat...	RDL File
 Student Course Lesson Status - Pie.rdl	RDL File
 Student Certificates.rdl	RDL File
 Sessions Per Course and Class.rdl	RDL File
 Sessions Per Course and Class - Pie.rdl	RDL File
 Organizations Per Class.rdl	RDL File
 Organizations Per Class - Pie.rdl	RDL File
 Lesson Time Per Course and Class.rdl	RDL File
 Divisions Per Class.rdl	RDL File
 Divisions Per Class - Pie.rdl	RDL File
 Courses Per Class.rdl	RDL File
 Courses Per Class - Pie.rdl	RDL File
 trackernet.gif	GIF File
 certificateclass.gif	GIF File

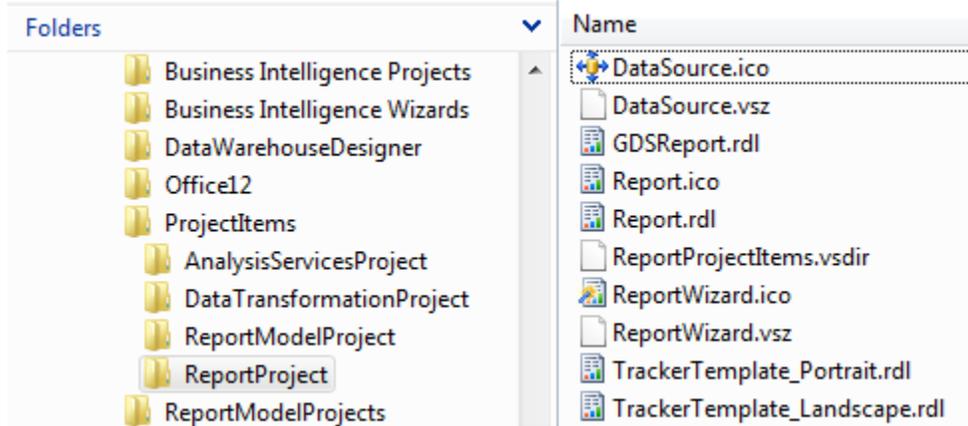
Documentation

In addition to the report and chart directories, the *Documentation* directory contains this Word file and its PDF equivalent.

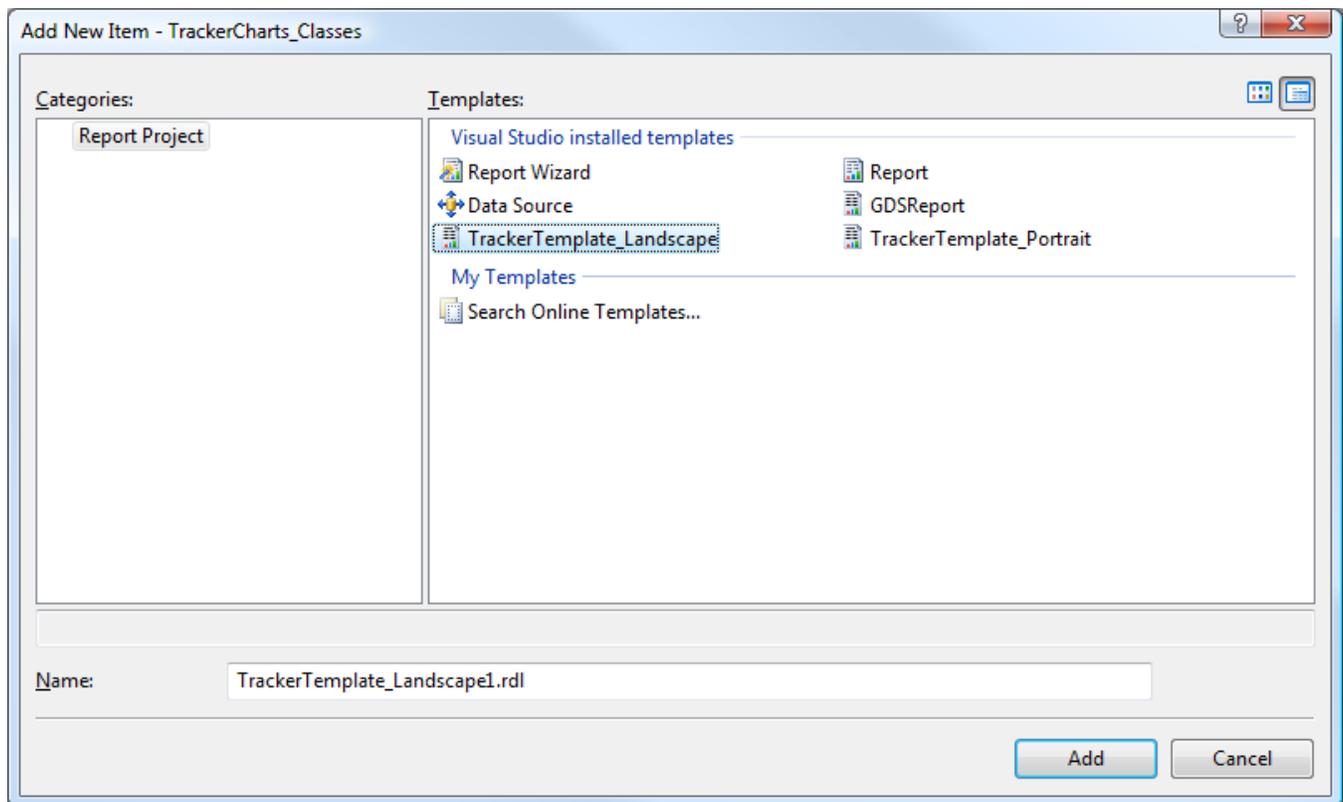
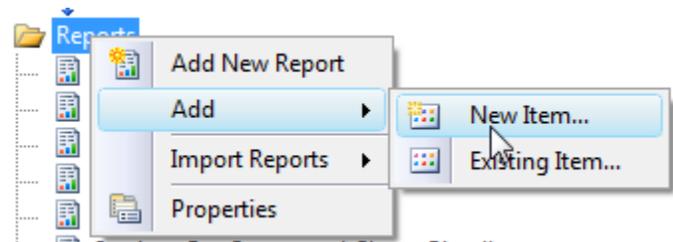
Templates

The *TrackerReports_Templates* contains the portrait and landscape report templates used for these reports and charts. In most cases, those of you who want to create new reports will copy and paste an

existing report. However, if you want to create new reports based on these templates, copy *TrackerTemplate_Landscape.rdl* and *TrackerTemplate_Portrait.rdl* to C:\Program Files\Microsoft Visual Studio 9.0\Common7\IDE\PrivateAssemblies\ProjectItems\ReportProject³. This is shown below.



Once you have copied the files to this location, you will have these additional choices when you create a new report as shown to the right. The resulting dialog box is displayed below.



³ This is for Visual Studio 2008 on a 32-bit operating system. Adjust accordingly for your version of Visual Studio.

Model

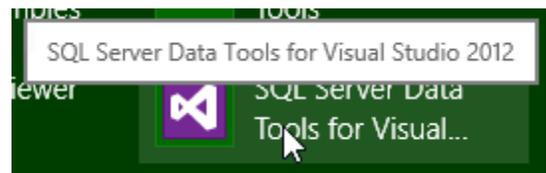
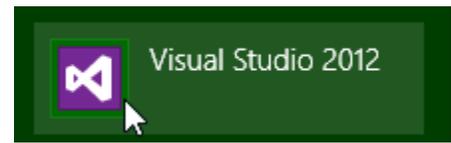
The *TrackerReportsModel* directory contains the *TrackerReportsModel* data model solution used in ad-hoc reporting. We will cover the use of the model and the associated *Report Builder* application later in this document.

Deploying Reports

You have two options for deploying your reports and charts. Using the first option, Visual Studio, is normally preferable since it is considerably faster – you can deploy each solution as a whole rather than having to import each individual report or graph.

Visual Studio/SQL Server Data Tools for Visual Studio Deployment

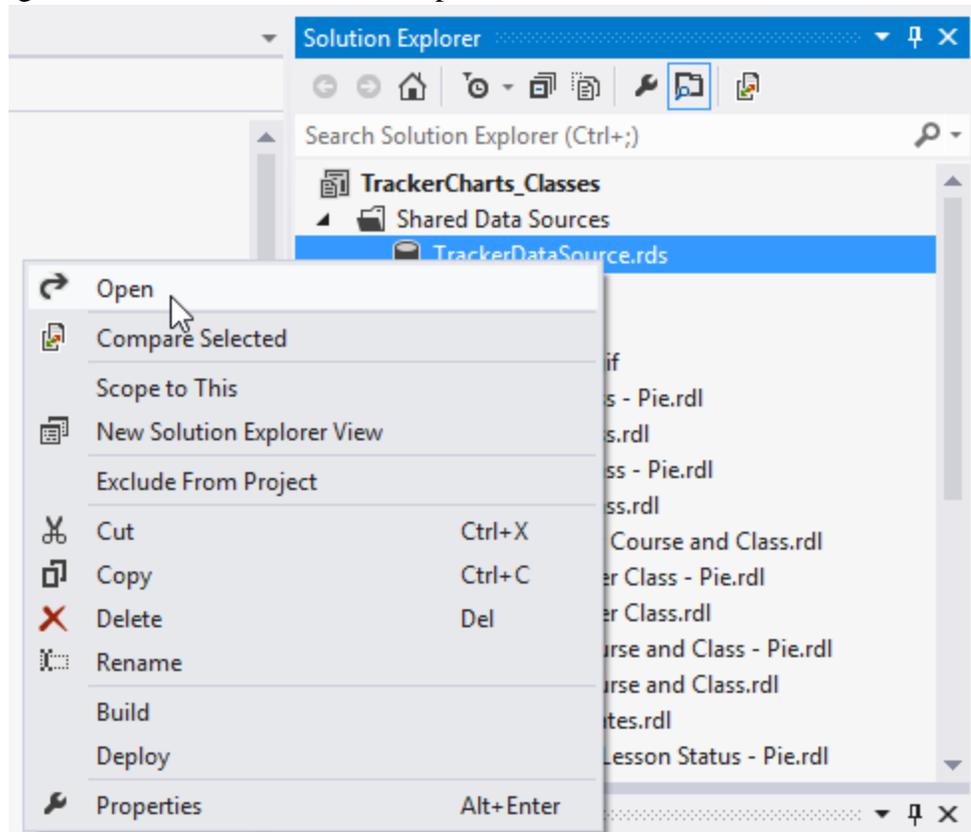
You can use the full *Visual Studio* or the more specialized *SQL Server Data Tools for Visual Studio* to create, edit, and deploy reports. You launch either one from the Start menu as shown to the right. You can then open each of the Solutions mentioned in the previous section. We break out the different type of reports and charts in separate solutions to make it easier to deploy them to their own folders with Reporting Services.

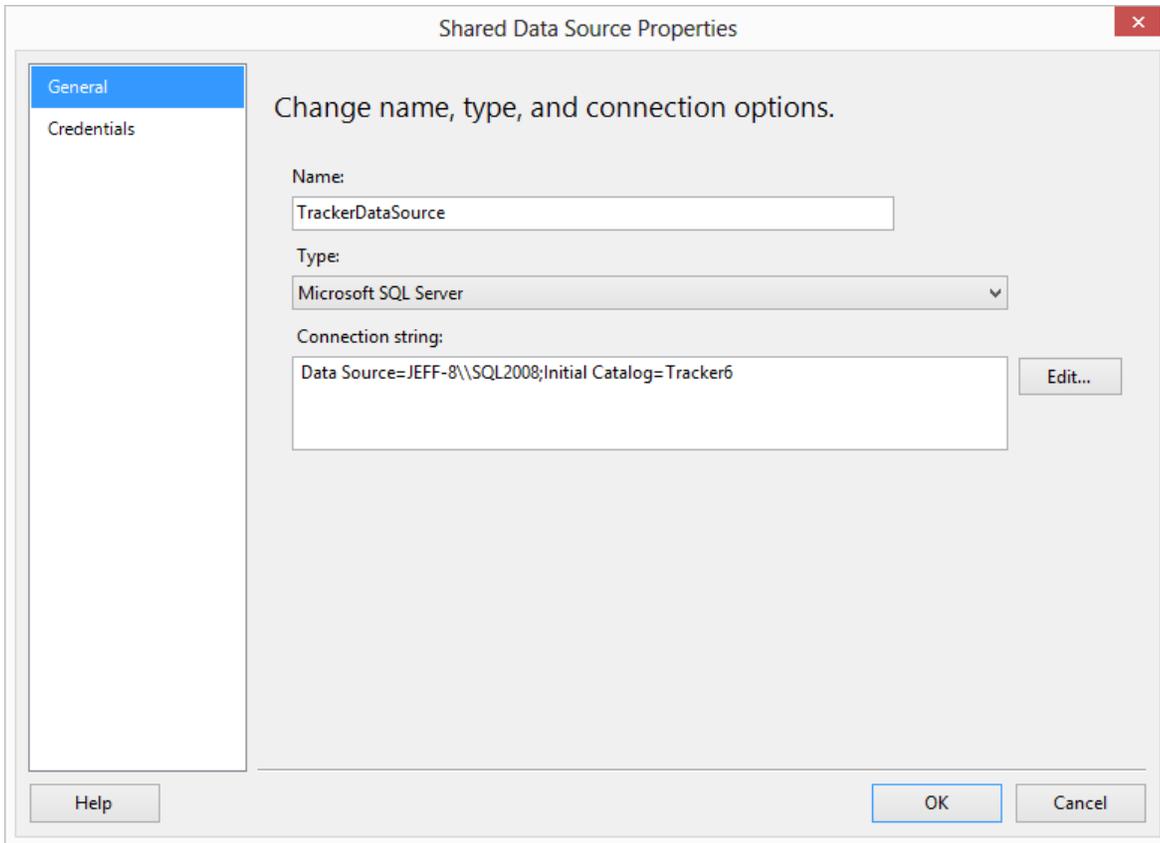


Data Source

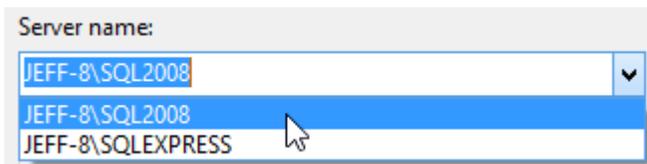
For the first solution you edit, you will likely want to customize the data source. This is particularly true if you want to preview the report while within Visual Studio. Otherwise, you can wait to set up the data source from within Reporting Services. We will cover that aspect later in this document.

To edit the data source, open the *TrackerDataSource.rds* file from within *Solution Explorer* as shown to the right. Note that this is a “Shared Data Source.” This means that all the reports in the solution share this one data source. Once you select “Open,” you will then see a dialog like the one below.

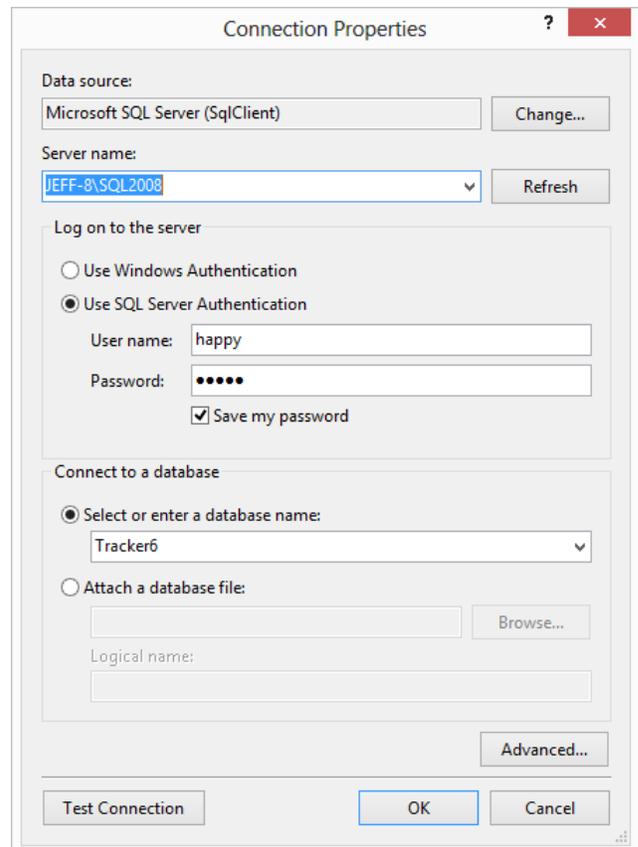




You will keep the *Type* as “Microsoft SQL Server,” but you may want to change the “Connection string.” To do that, click the “Edit...” button. This will show a screen like the one to the right. If you are connecting to a SQL Server instance on your local machine, you can select the instance from the list, as shown below.



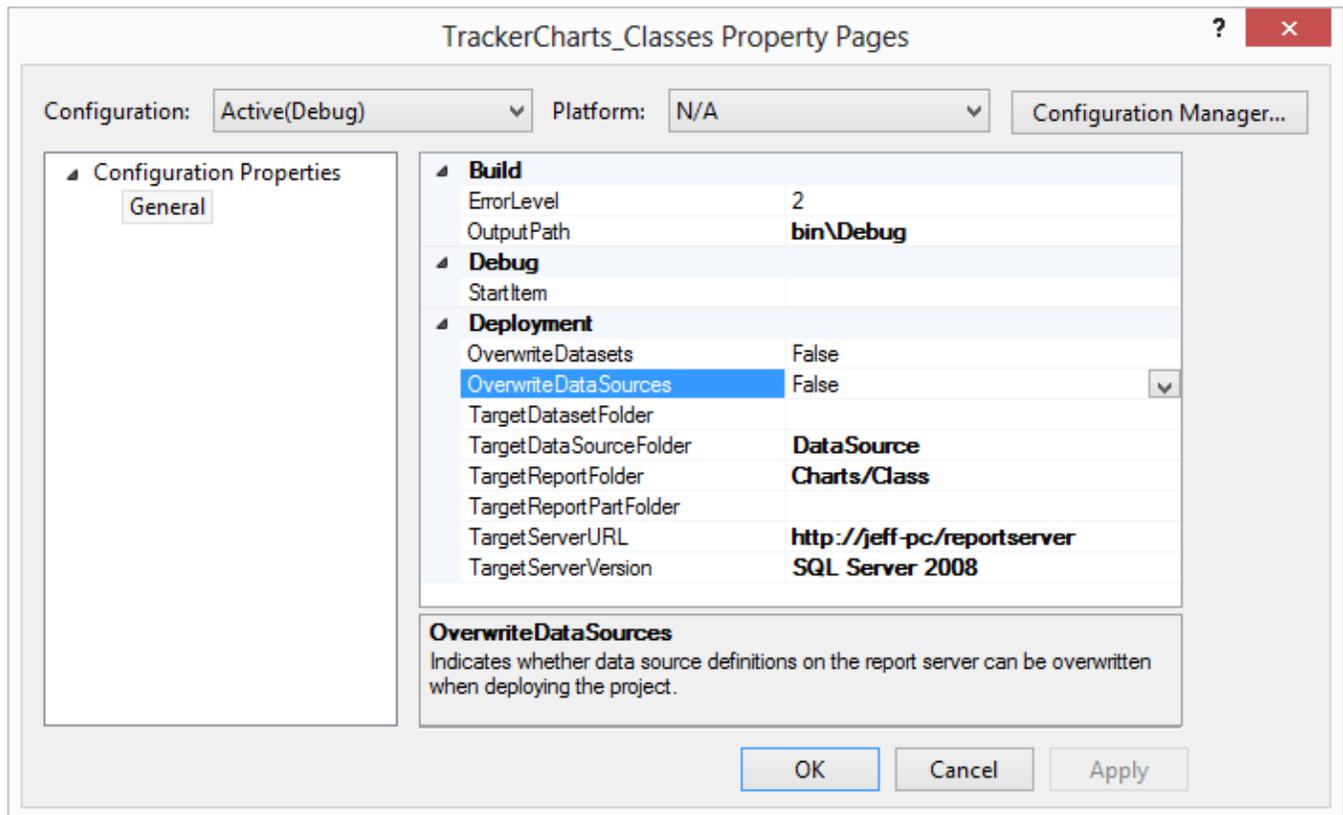
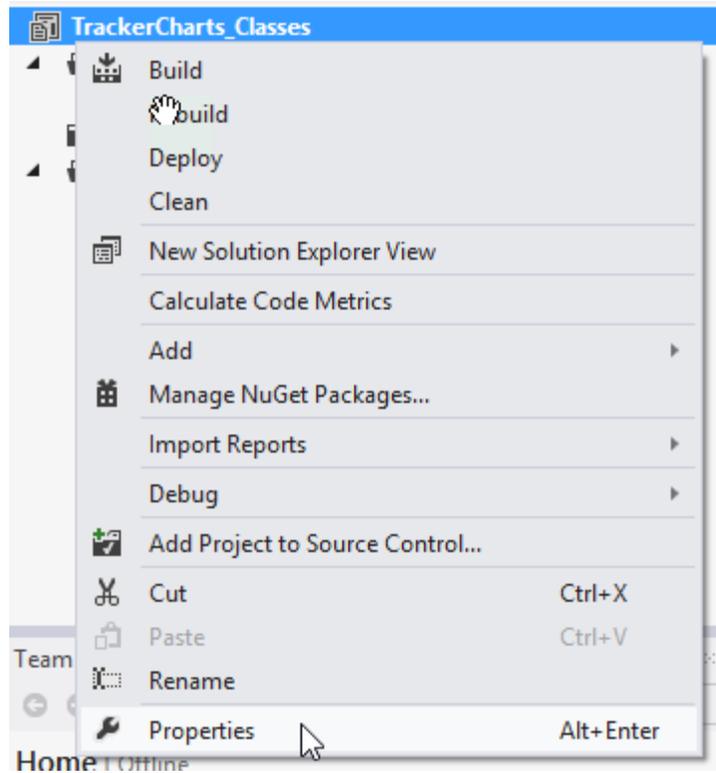
You can use either Windows or SQL Server authentication. Once that is set, you can select the database from the list. It is a good idea to click the “Test Connection” button to make sure everything is working. Note that you can select versions of SQL Server earlier than 2008 for testing from within Visual Studio but need to use 2008 or later once you connect to a data source from within Reporting Services. Once you close this dialog, you don’t need



to worry about the “Credentials” tab in the original dialog – that tab also shows the authentication information.

Deployment Properties

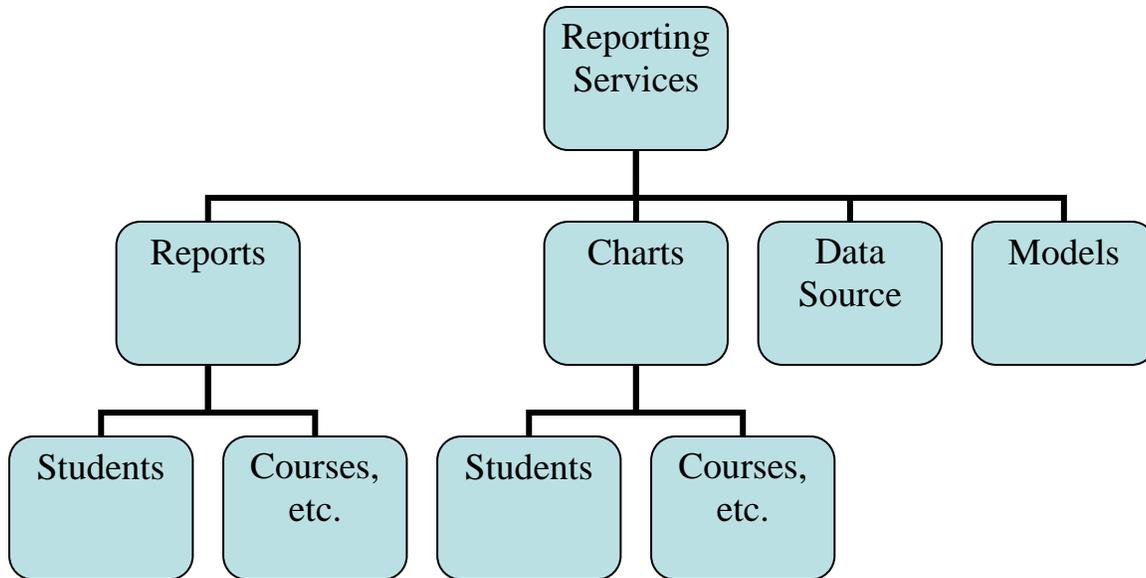
You next want to edit the properties of the project⁴ itself. To do that, right-click on the project and select “Properties” as shown to the right. You then see this dialog box:



⁴ In Visual Studio parlance, the core unit is a project. A *solution* is then a collection of one or more projects. We talked about solution files earlier as that is the primary way you open up the solution. In our case, each of our solutions only has a single project. So in this document, solution and project are largely interchangeable.

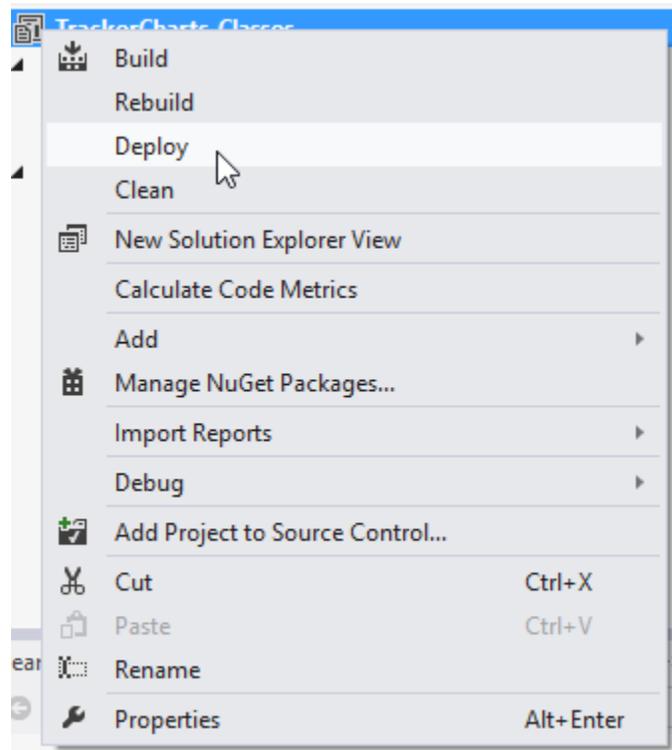
The dialog above is configured for deployment to a local instance for testing. The values will be somewhat different if you are deploying straight to your web server, as we will show shortly. Let's look at each of the key settings in turn.

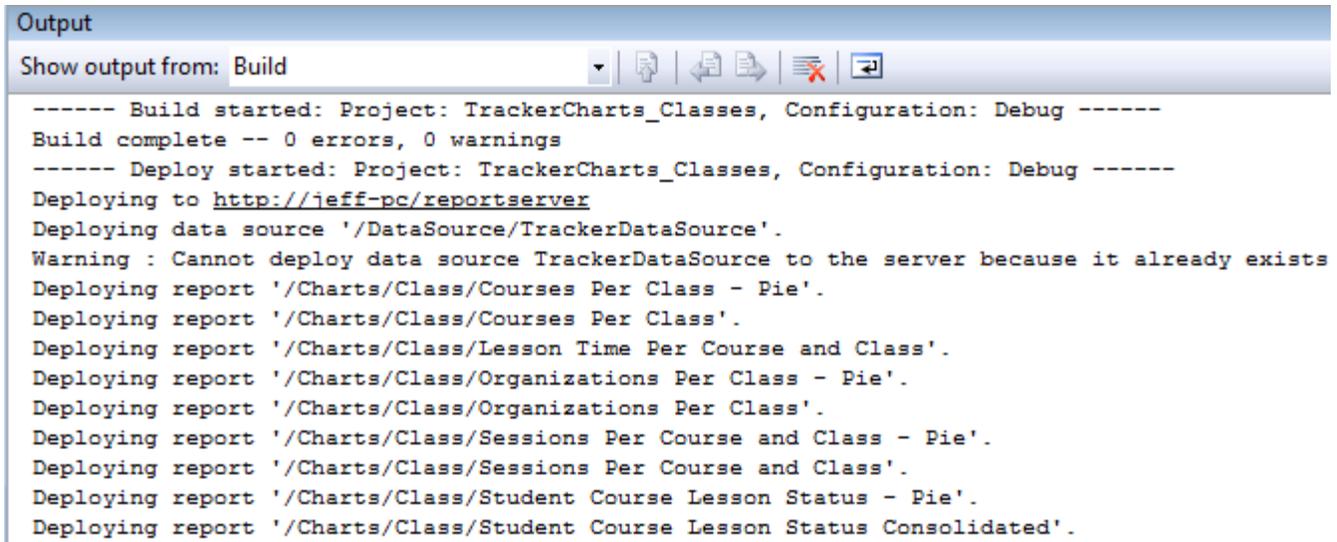
<i>Setting</i>	<i>Description</i>
OverwriteDataSources	This setting determines whether the deployment process overwrites an existing data source on your Reporting Services server. You typically want to leave this a <i>False</i> so that you can have a different data source locally when testing than what you use on the production server.
TargetDataSourceFolder	<p>This is the folder within Reporting Services where you want to locate the data source. You normally want this to be a single folder which is the same for all of your solutions. That allows you to share that data source. It also keeps you from having to manually set the data source for each report once you deploy it, saving you quite a bit of time.</p> <p>If you are going straight to your web server, you may have a more complete path. For example, when we deploy directly to our Reporting Services application hosted by DiscountASP.NET, we use this value for the <i>TargetDataSourceFolder</i>: /plattecany1/reports/DataSource</p>
TargetReportFolder	<p>This is the folder within Reporting Services where you want to locate the reports/charts in the current solution. We recommend a layout like the chart on the next page. One advantage of separating items into different folders is that you can set different permissions for each folder. However, you are welcome to organize the reports in a way that best fits your organization.</p> <p>If you are going straight to your web server, you may have a more complete path. For example, when we deploy directly to our Reporting Services application hosted by DiscountASP.NET, we use a value like this for the <i>TargetReportFolder</i>: /plattecany1/reports/Charts/Class</p>
TargetReportFolder	<p>This is the folder within Reporting Services where you want to locate the reports/charts in the current solution. We recommend a layout like the chart on the next page. One advantage of separating items into different folders is that you can set different permissions for each folder. However, you are welcome to organize the reports in a way that best fits your organization.</p> <p>If you are going straight to your web server, you may have a more complete path. For example, when we deploy directly to our Reporting Services application hosted by DiscountASP.NET, we use a value like this for the <i>TargetReportFolder</i>: /plattecany1/reports/Charts/Class</p>



Completing the Deployment

Once you have set all the properties to their desired values, right-click on the project and select the “Deploy” option as shown to the right. You will then see the results in the Visual Studio Output Window. This is shown below. Note that the warning about not copying the Data Source is normal once you have deployed it once. This is what the “False” value for *OverwriteDataSource* means.

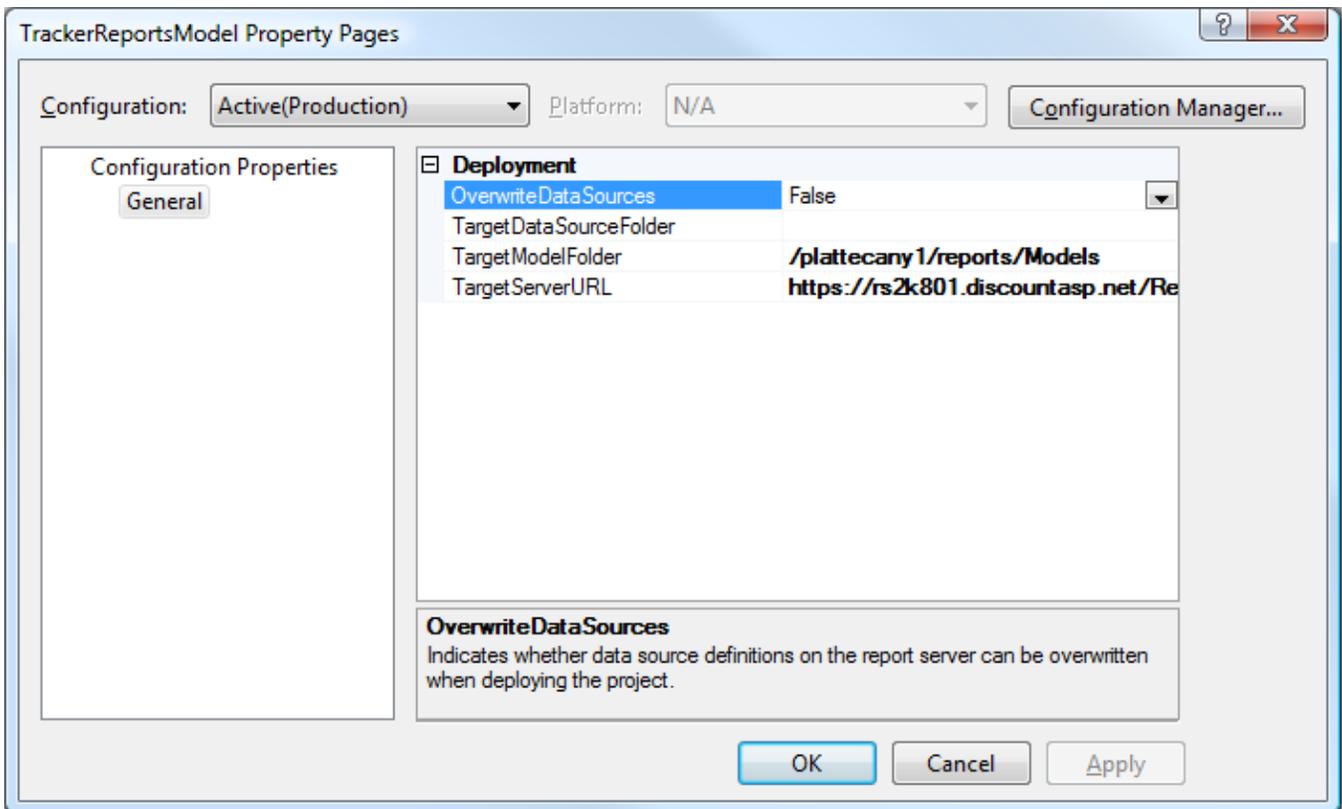




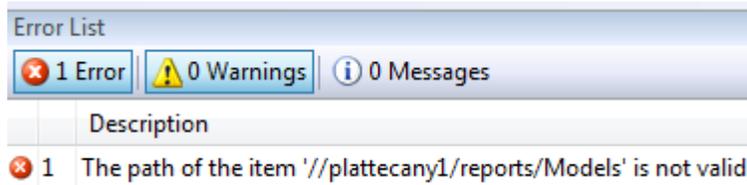
```
----- Build started: Project: TrackerCharts_Classes, Configuration: Debug -----
Build complete -- 0 errors, 0 warnings
----- Deploy started: Project: TrackerCharts_Classes, Configuration: Debug -----
Deploying to http://jeff-pc/reportserver
Deploying data source '/DataSource/TrackerDataSource'.
Warning : Cannot deploy data source TrackerDataSource to the server because it already exists
Deploying report '/Charts/Class/Courses Per Class - Pie'.
Deploying report '/Charts/Class/Courses Per Class'.
Deploying report '/Charts/Class/Lesson Time Per Course and Class'.
Deploying report '/Charts/Class/Organizations Per Class - Pie'.
Deploying report '/Charts/Class/Organizations Per Class'.
Deploying report '/Charts/Class/Sessions Per Course and Class - Pie'.
Deploying report '/Charts/Class/Sessions Per Course and Class'.
Deploying report '/Charts/Class/Student Course Lesson Status - Pie'.
Deploying report '/Charts/Class/Student Course Lesson Status Consolidated'.
```

Model Deployment

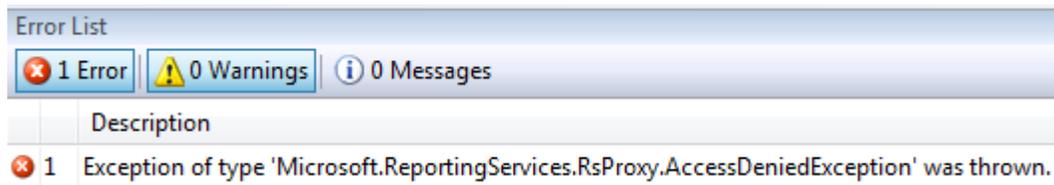
You can use the same approach for deploying the *TrackerModel* for ad-hoc reporting if you have sufficient rights. The first thing to watch is that the version of Reporting Services that comes with *SQL Server 2008 Express* does not support models. Our shared hosting Reporting Services provider does support them, but we are not able to upload the model directly due to their security setting. This is apparently a common issue. Here is a screen capture of our project properties:



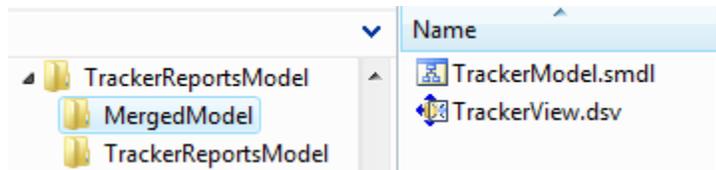
Trying to deploy gives this error message:



The next recommended step is to use a complete URL for the *TargetModelFolder*, such as <https://rs2k801.discountasp.net/ReportServer/plattecany1/reports/Models>. Unfortunately, that gives this security error.



Luckily, not all is lost. The answer is to import the files directly from *Report Manager* as described in the next section. However, it is important to use the *TrackerModel.smdl* file from the *MergedModel* directory rather than the one in the *TrackerReportsModel* directory as shown to the right. This is needed since modifications needed to be made to the file in order to be accepted directly into *Report Manager*.

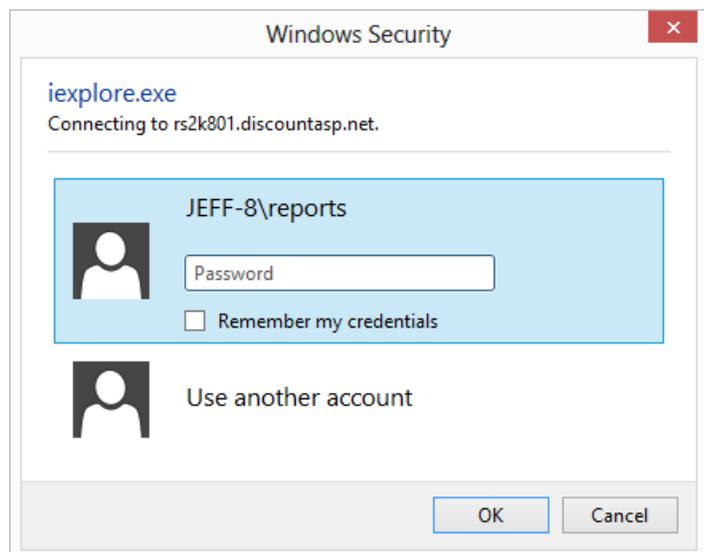


Import Files from Report Manager

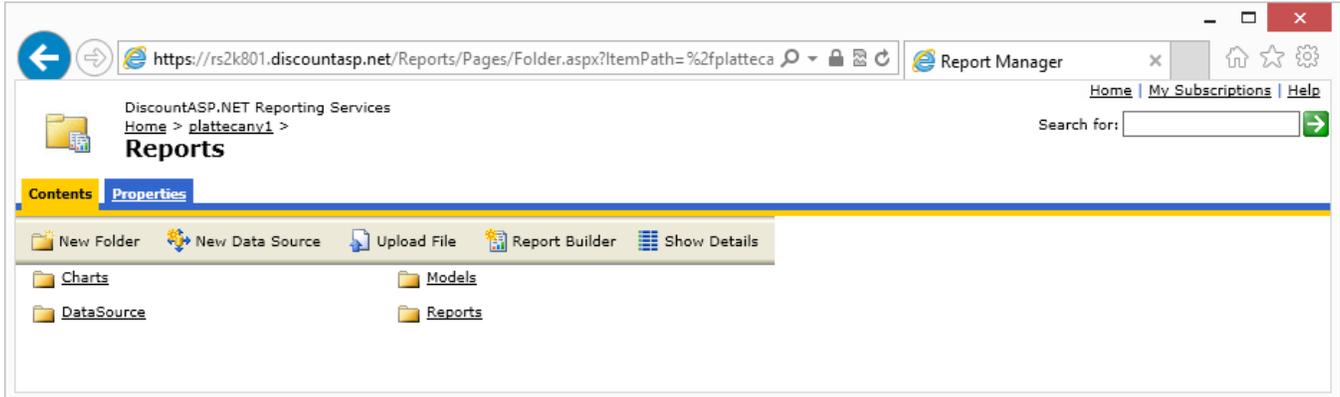
You and your users will log into *Report Manager* to actually view reports. You can also use it to deploy reports in the first place or change their values later.

Logging into Report Manager

The first step is to log in. The URL for your local machine will be <http://localhost/Reports> or <http://<machine name>/Reports>. For your server, it will be something like <https://rs2k801.discountasp.net/Reports>. Unless you have done special programming, *Report Manager* uses Windows Authentication. In an Enterprise environment, this means that you may not be prompted for a username and password once you are already logged into your network. Otherwise, you will see a dialog box like the one to the right.



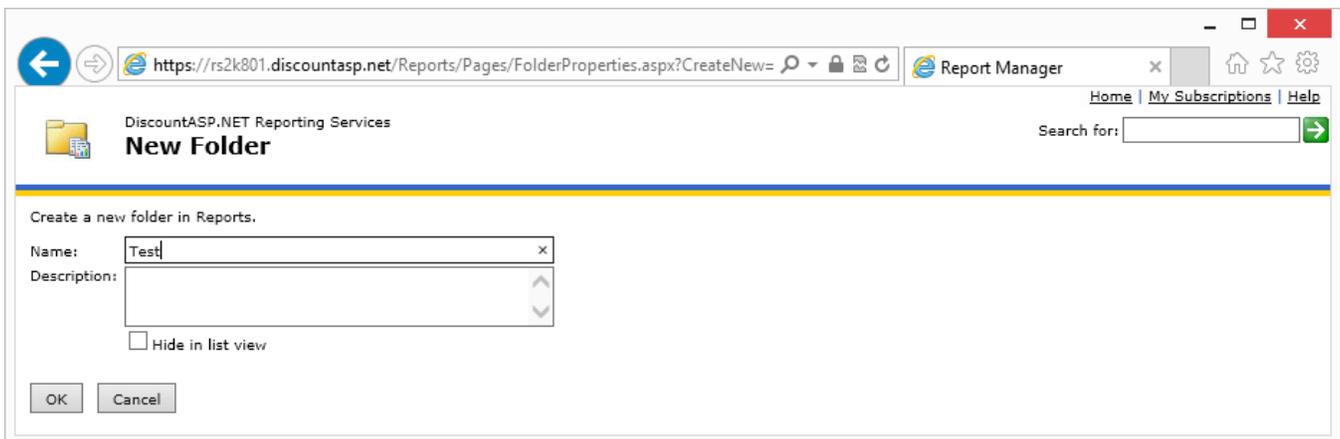
The *Report Manager* interface is shown below.



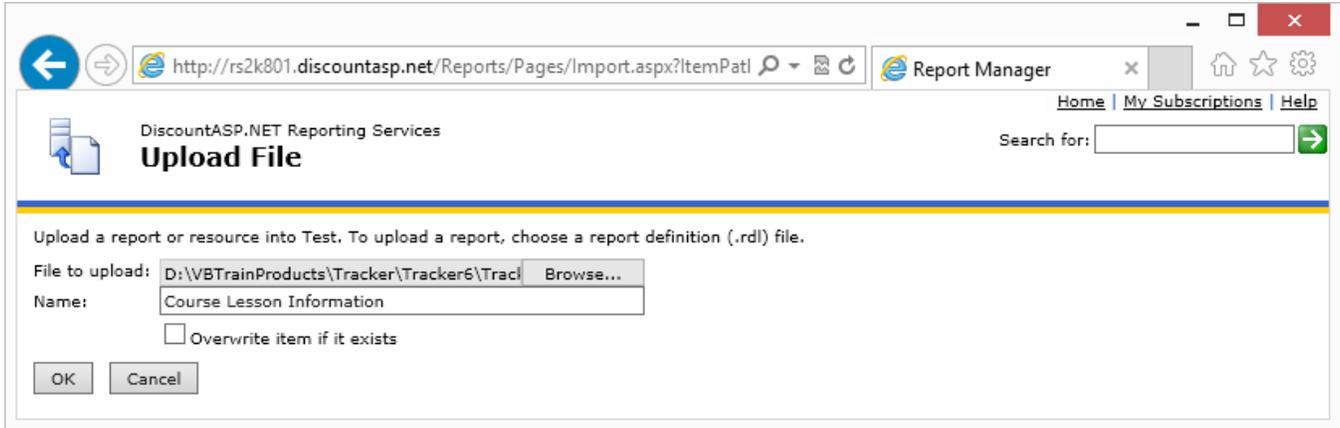
Uploading Individual Reports/Charts

Let's go through an example of adding a report to a new "Test" directory.

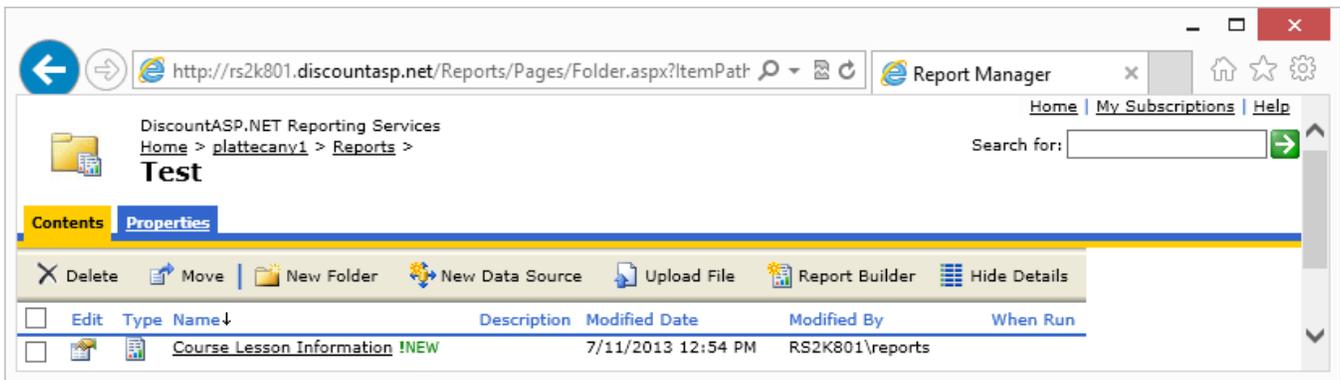
1. Our first step is to click the "New Folder" button on the toolbar. We'll enter "Test" for the name as shown below.



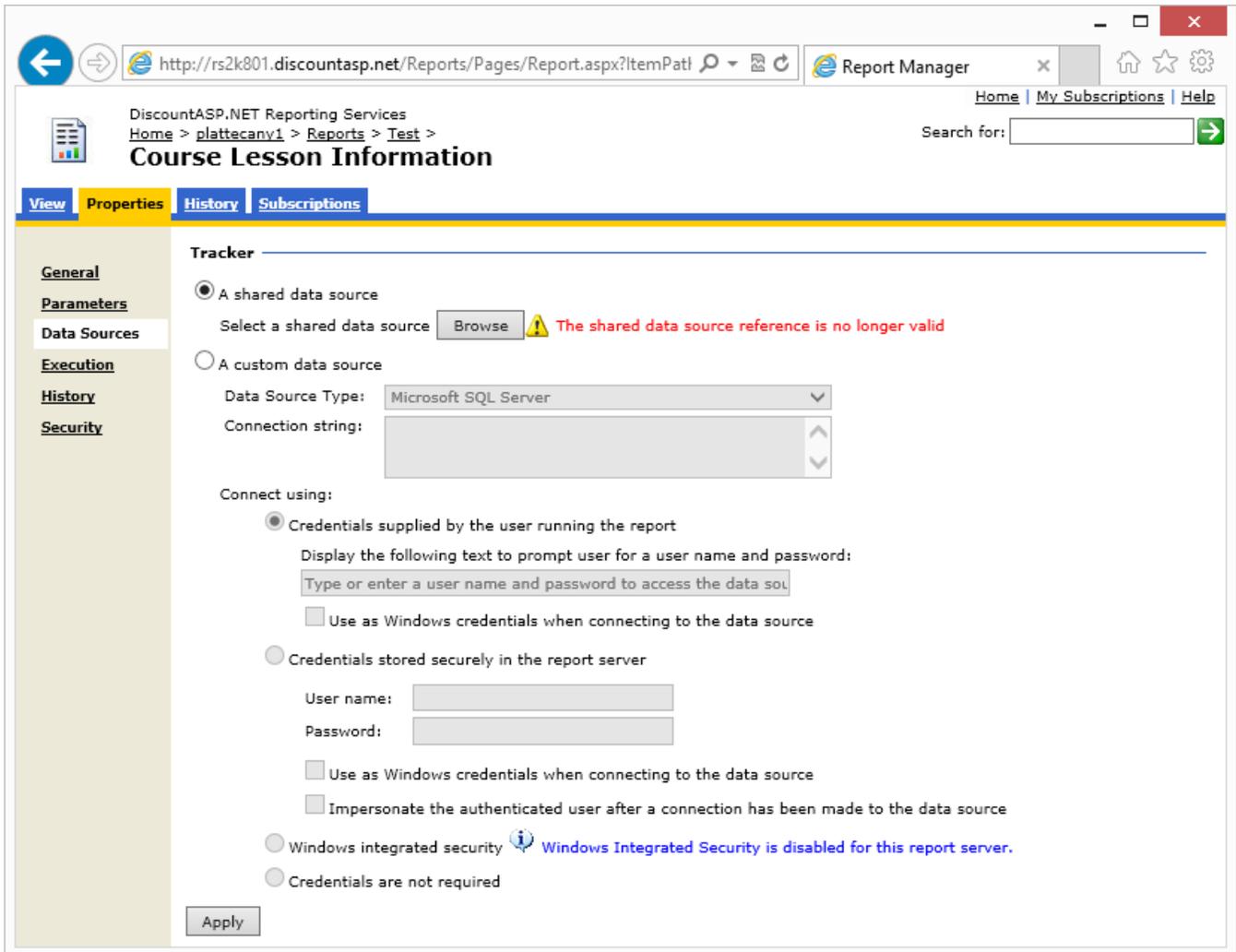
2. We navigate to that folder and click the "Upload File" button on the toolbar. We then click the "Browse..." button to navigate to the desired *.rdl file on our local machine. In the screen capture below, we chose "Course Lesson Information.rdl" from the *TrackerReports_Students* directory.



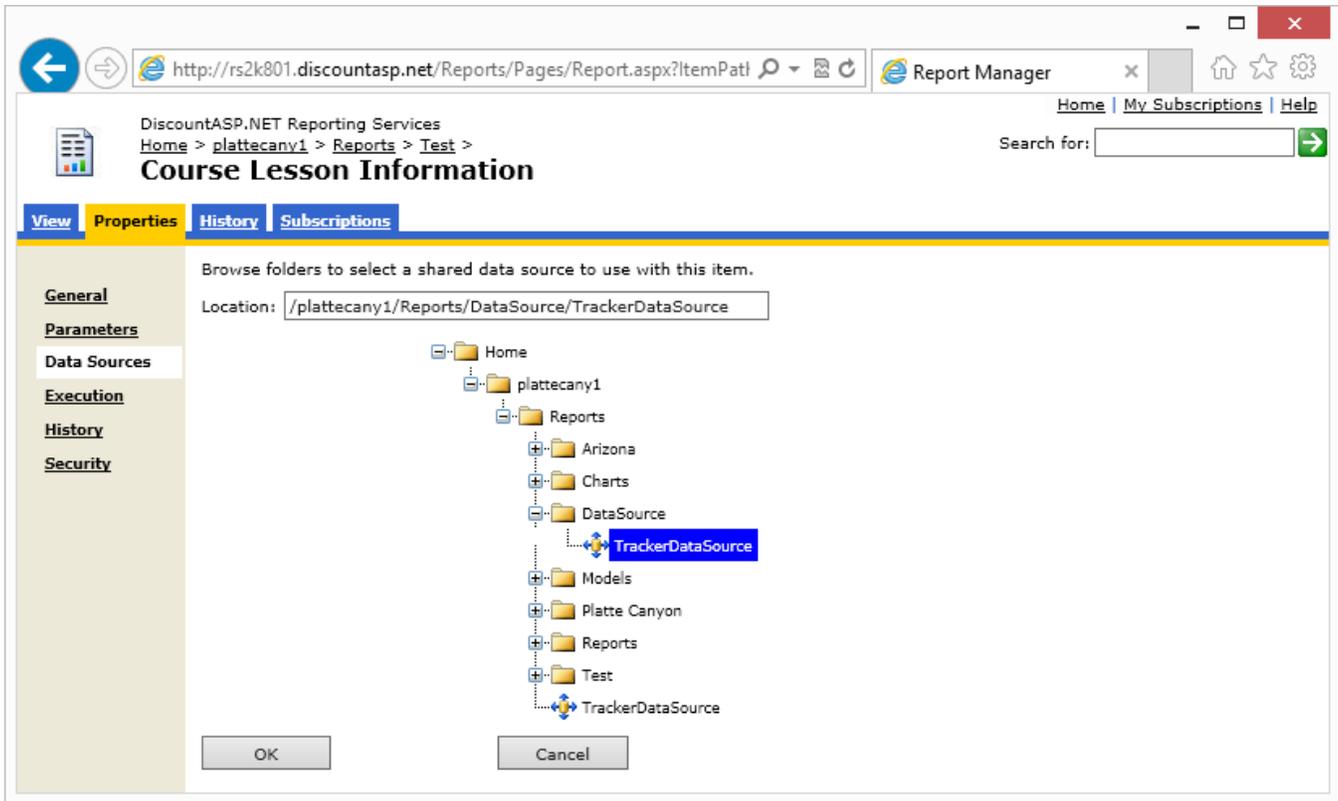
3. We may get an initial error message, but you can ignore it for now. We then navigate to the *Test* directory and click the “Show Details” button as shown below.



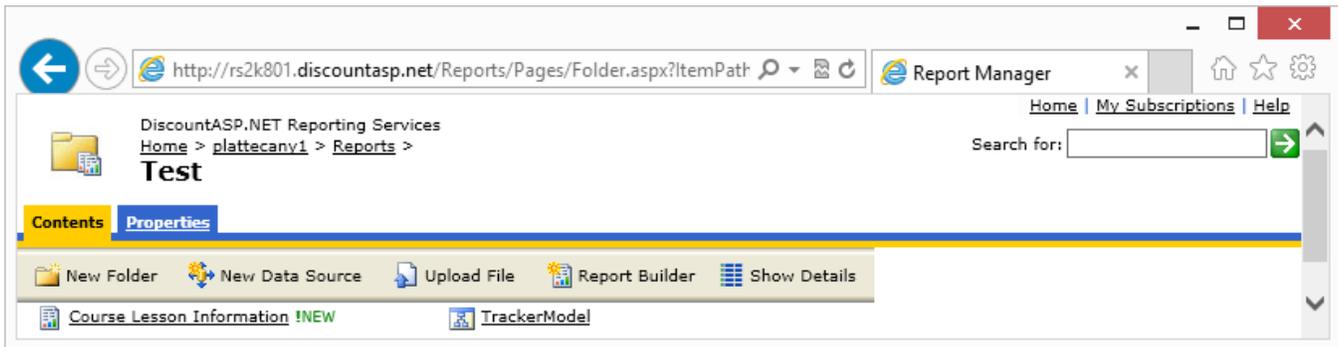
4. We next click on the properties button in the “Edit” column. On the left side, click on “Data Sources.” You will then see a screen like the one below.



5. We will cover creating a Data Source in the next section. You typically do NOT want to use a custom data source but rather a shared data source. Otherwise, you'll have multiple data sources all pointing to the same database. Since we already have a data source on the server, we click the "Browse" button to select it. This is shown below.

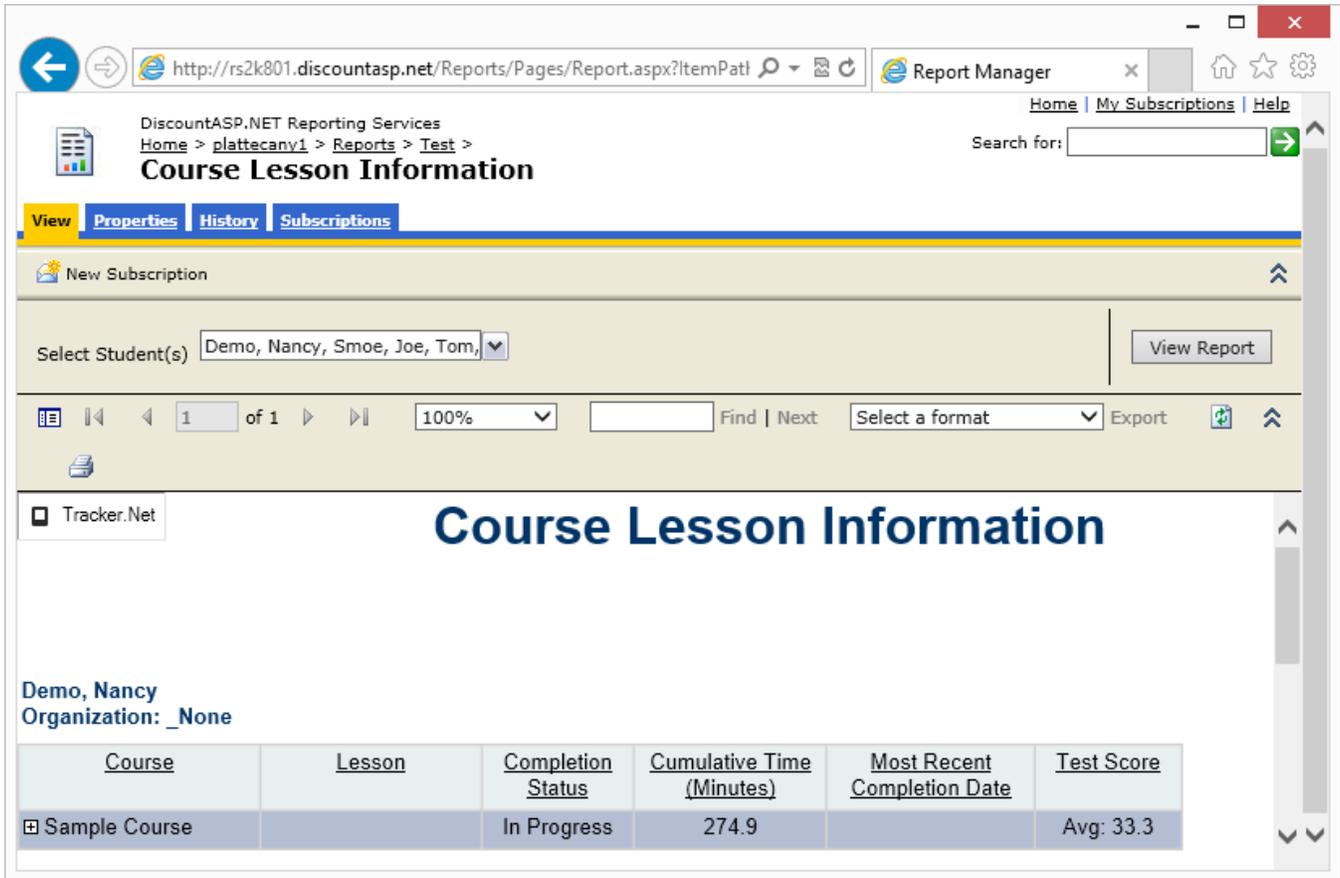


6. We navigate to the *DataSource* directory and select *TrackerDataSource*. We will show how to configure this source in the next section. Click the OK button and then the Apply button to save your changes⁵. Navigate to the Test directory and click on the report.

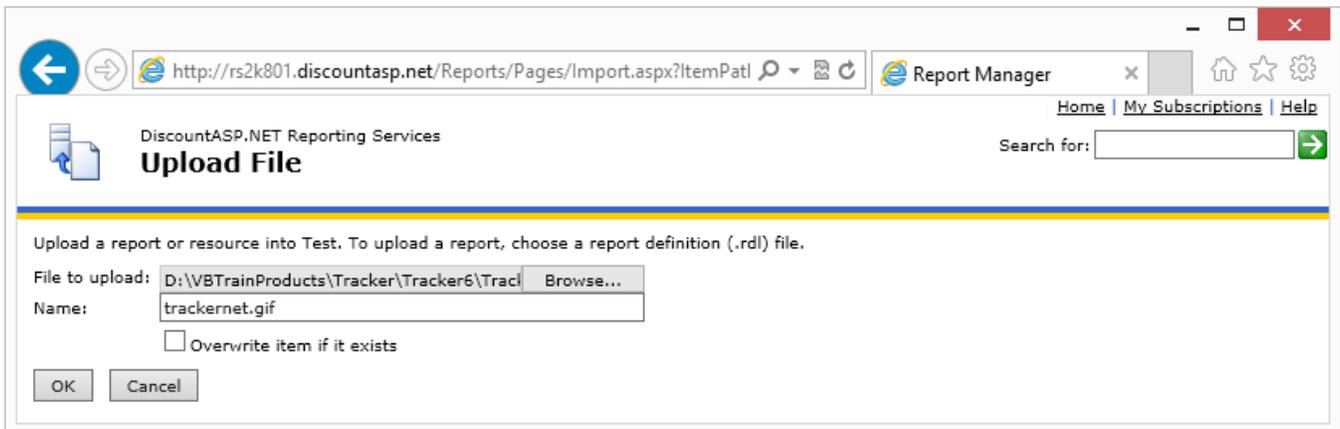


7. We get some results but have a bad graphic link at the upper left. This is because we did not upload trackernet.gif to our *Test* directory. Note that you can take your own logo and name it trackernet.gif and add that so that the reports show your logo instead of ours.

⁵ I have forgotten to click the *Apply* button several times and lost my edits. Be sure to scroll down the window and look for the *Apply* button.



8. We use the same “Upload File” button as before to add the graphic file.



Since we need to go through this process for each of our reports/graphs⁶, you can see why we recommend deploying via Visual Studio if possible.

Data Source

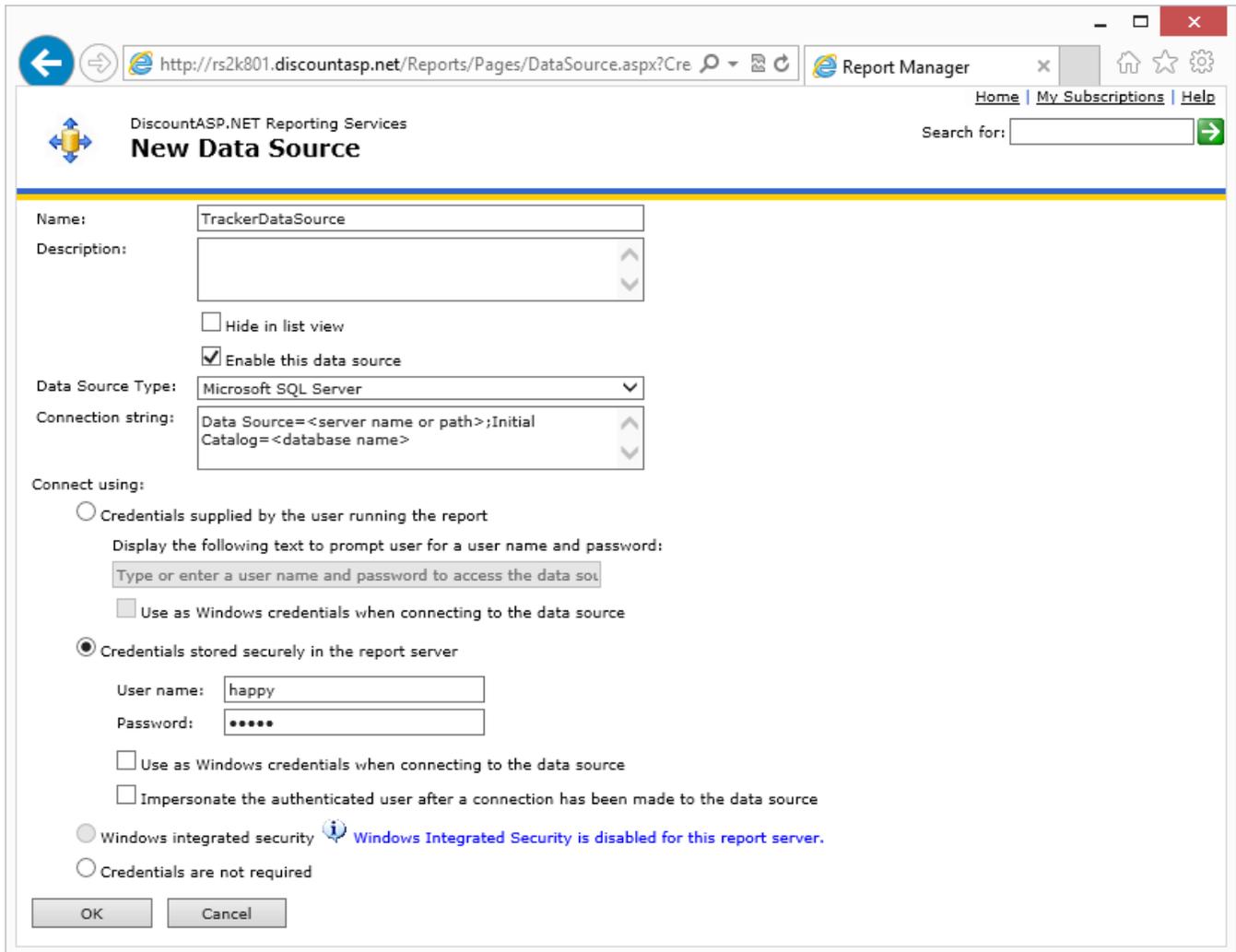
Adding a data source is a matter of knowing the important connection information:

⁶ Except that you only need to upload trackernet.gif once per directory.

Tracker Reports Documentation

- Server name/path
- Database name
- Username
- Password

To get started, create a folder as in the previous section or navigate to the desired folder. Then click the “New Data Source” button on the toolbar. The resulting screen is shown below:



The screenshot shows a web browser window with the URL `http://rs2k801.discountasp.net/Reports/Pages/DataSource.aspx?Cre`. The page title is "DiscountASP.NET Reporting Services" and the main heading is "New Data Source". The browser's address bar shows "Report Manager" and navigation icons. The page has a search bar and links for "Home", "My Subscriptions", and "Help".

The "New Data Source" form contains the following fields and options:

- Name:** TrackerDataSource
- Description:** (empty text area)
- Hide in list view
- Enable this data source
- Data Source Type:** Microsoft SQL Server
- Connection string:** Data Source=<server name or path>;Initial Catalog=<database name>
- Connect using:**
 - Credentials supplied by the user running the report
 - Display the following text to prompt user for a user name and password:
Type or enter a user name and password to access the data source
 - Use as Windows credentials when connecting to the data source
 - Credentials stored securely in the report server
 - User name: happy
 - Password: (masked with dots)
 - Use as Windows credentials when connecting to the data source
 - Impersonate the authenticated user after a connection has been made to the data source
 - Windows integrated security  Windows Integrated Security is disabled for this report server.
 - Credentials are not required

At the bottom of the form are "OK" and "Cancel" buttons.

Note the format of the *Connection String*. You want to have *DataSource=* and *InitialCatalog=*. The *DataSource* is the name of the server. It can be something like (local) or a complete URL or IP address like `tcp:servername.discountasp.net`. Note that this is the same as the “server=” value in the “ConnectionString” key in your main Tracker.Net web.config file. The *InitialCatalog* is your database name. It will be something like Tracker. It is the same as the “database=” value in the Tracker.Net web.config file.

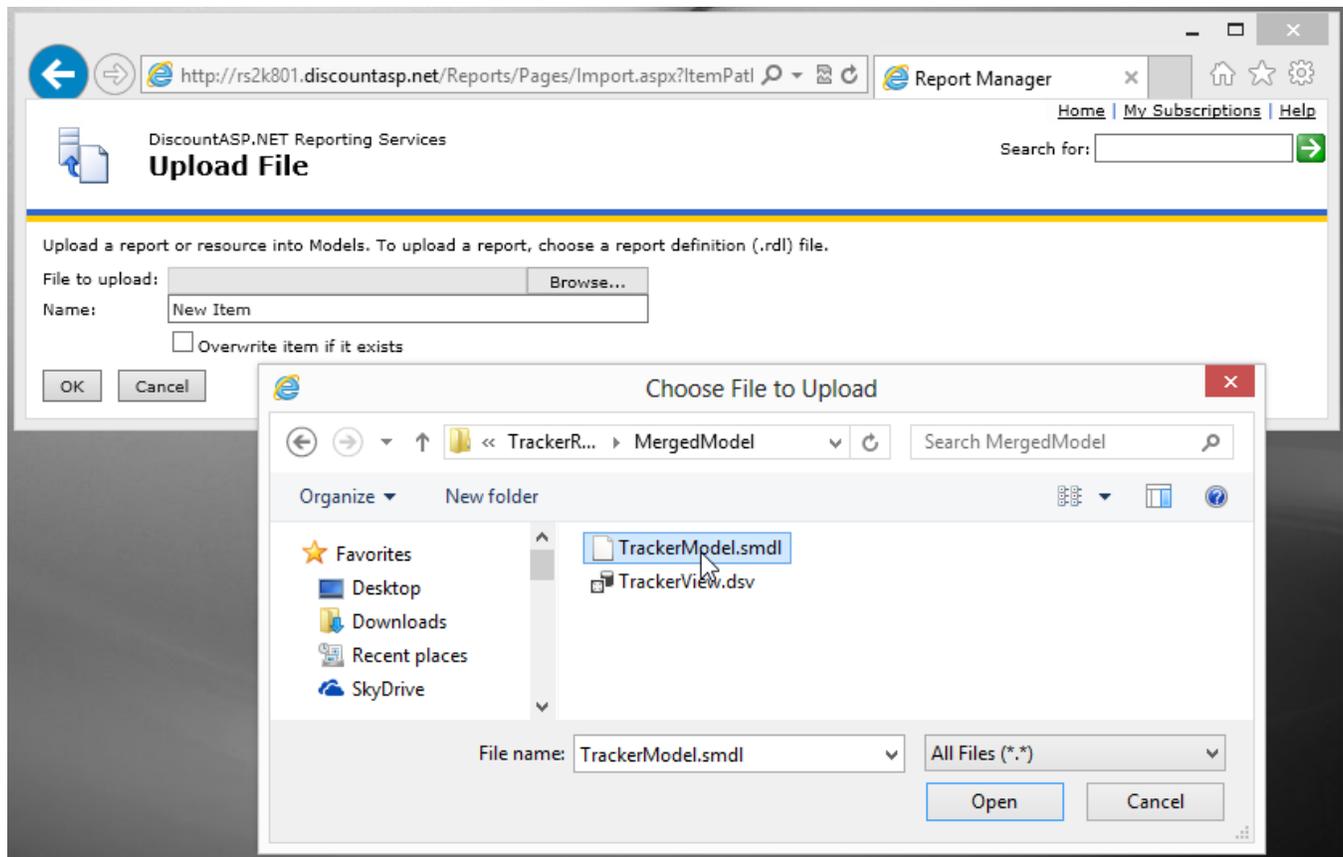
While you have the option of prompting the user for a username and password, that is typically not information that you want them to know. Instead, you can use Windows Authentication if available or store the credentials on the server. That is what we are doing in the screen capture above.

Click the “OK” button when you are finished. You can then assign this Data Source to individual reports as described in the previous section.

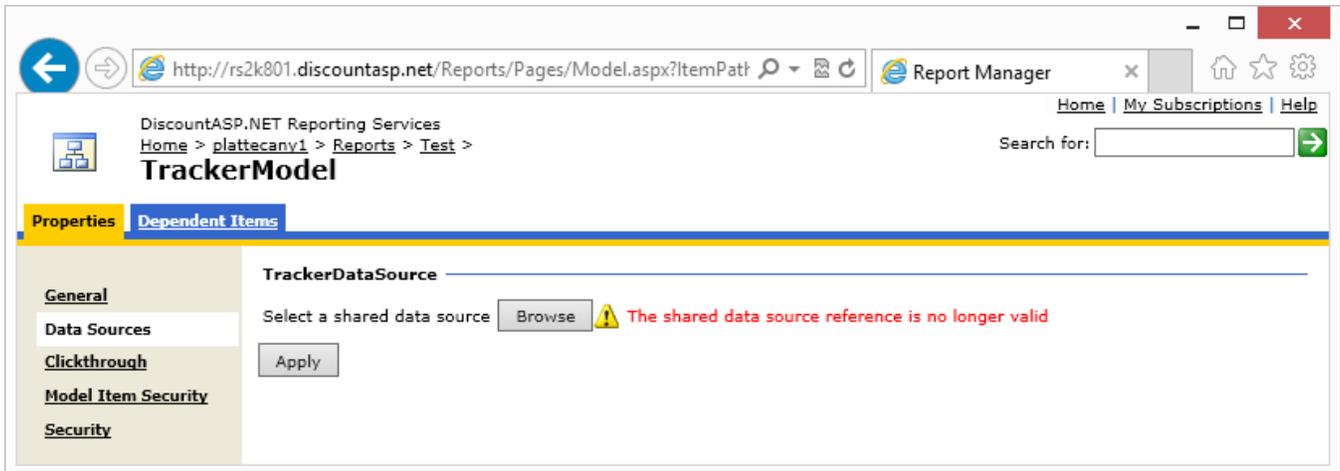
Model Deployment

As we mentioned earlier in the document, security considerations can make it difficult to deploy a model via Visual Studio. Here are the steps for doing it via *Report Manager*.

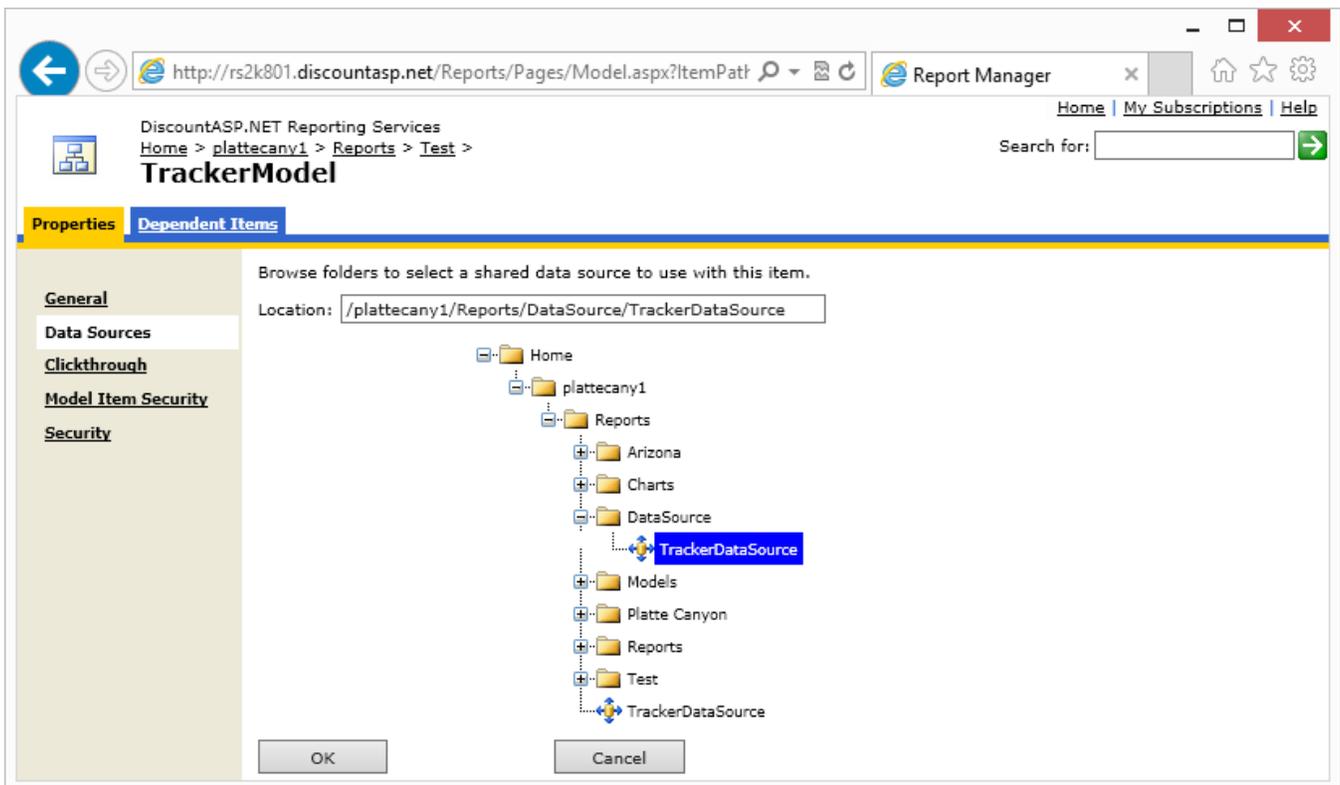
1. Navigate to the desired directory and click on the “Upload File” button on the toolbar.
2. Browse to the *TrackerModel.smdl* file in the *TrackerReportsModel\MergedModel* directory. This is shown below.



3. Click the OK button. Repeat the steps above and upload *TrackerView.dsv* to the same directory.
4. Click on *TrackerModel* and then on the *Data Sources* item on the left. The result is shown below.



5. Browse to your shared data source as shown below.



6. Click the OK button and then the Apply button.

We will cover use of the model later in this document. Notice that models are not available in the Reporting Services that comes with *SQL Server 2008 Express*.

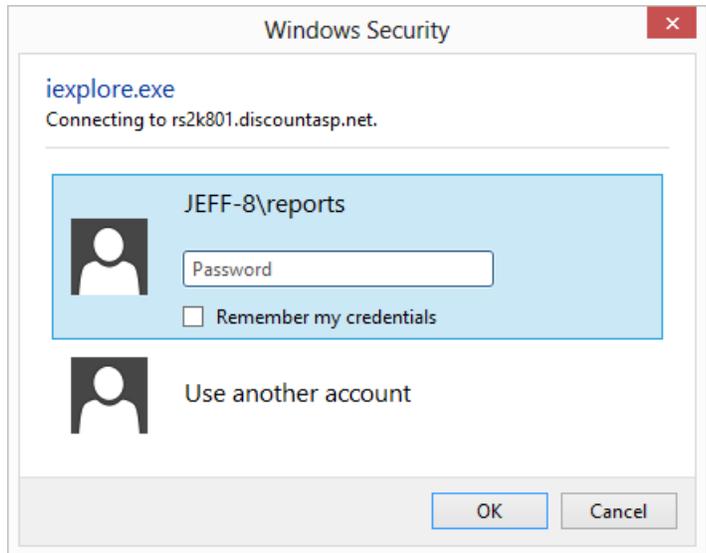
Accessing Reports and Charts

The primary way that users get access to your reports is by logging into *Report Manager*. Note that it is possible to limit which reports users are able to access. In addition, the ability to create linked reports

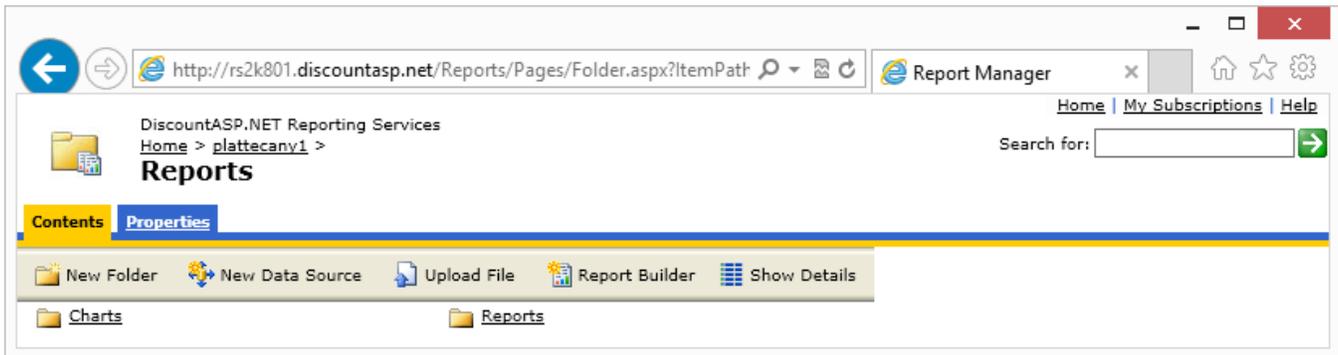
(covered in a later section) makes it possible to set different parameters for different users. For example, you can link *Organization* reports and set which organization can be viewed. You could then give a manager of that organization rights only to that linked report or to a directory of those reports.

Logging into Report Manager

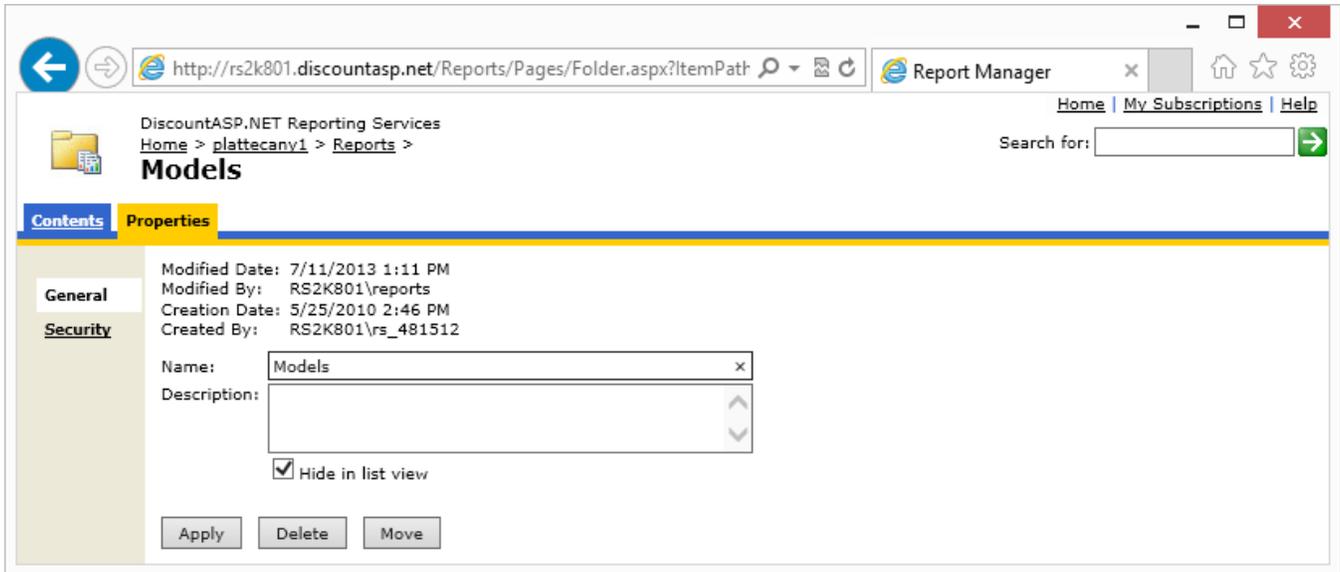
The first step is to log in. The URL for your local machine will be `http://localhost/Reports` or `http://<machine name>/Reports`. For your server, it will be something like `https://rs2k801.discountasp.net/Reports`. Unless you have done special programming, *Report Manager* uses Windows Authentication. In an Enterprise environment, this means that you may not be prompted for a username and password once you are already logged into your network. Otherwise, you will see a dialog box like the one to the right.



The *Report Manager* interface is shown below.

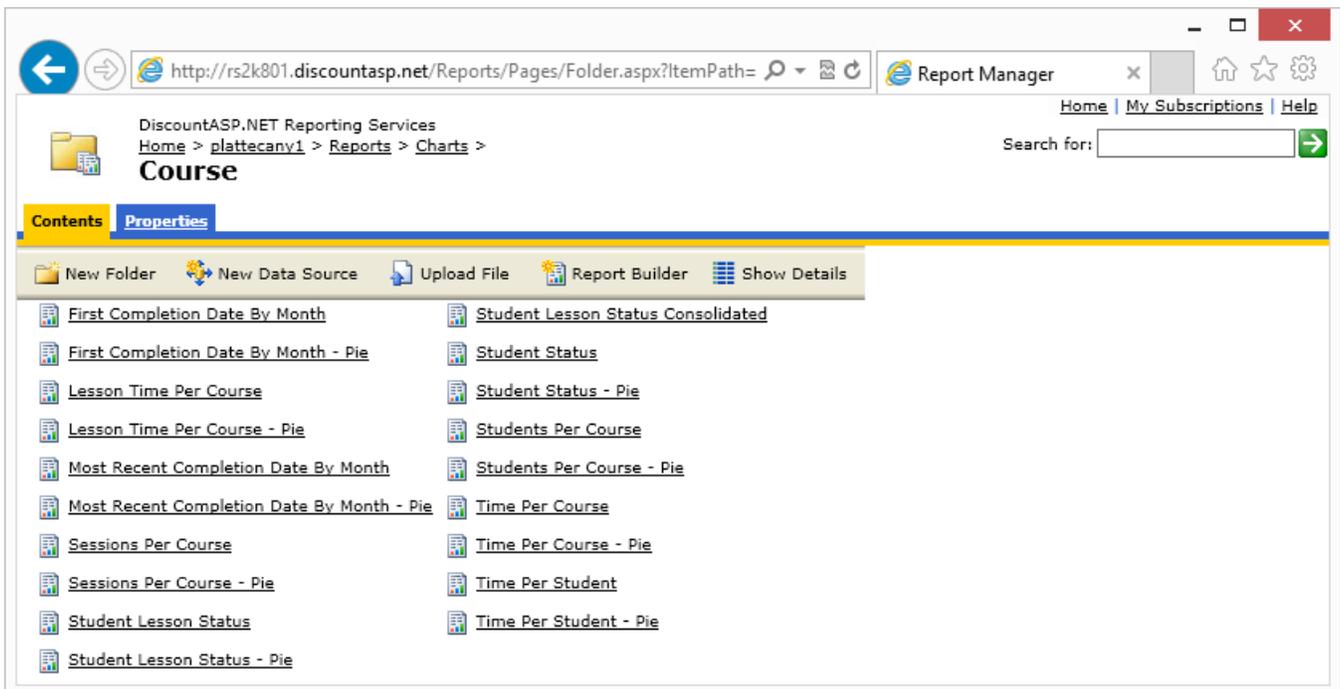


You may have noticed that some of the items like *DataSource* and *Models* that were in earlier screen captures are no longer present. This is because we have set them to be hidden in the "list view." You do this in the properties of the folder as shown below. You can also hide items like *trackernet.gif* within a folder.



Running Reports

To actually view a report or chart, go to either the *Reports* or *Charts* folder and then to the *Class*, *Course*, *Division*, *Lesson*, *Organization*, or *Student* folder. Doing this for *Charts – Course* is shown below.



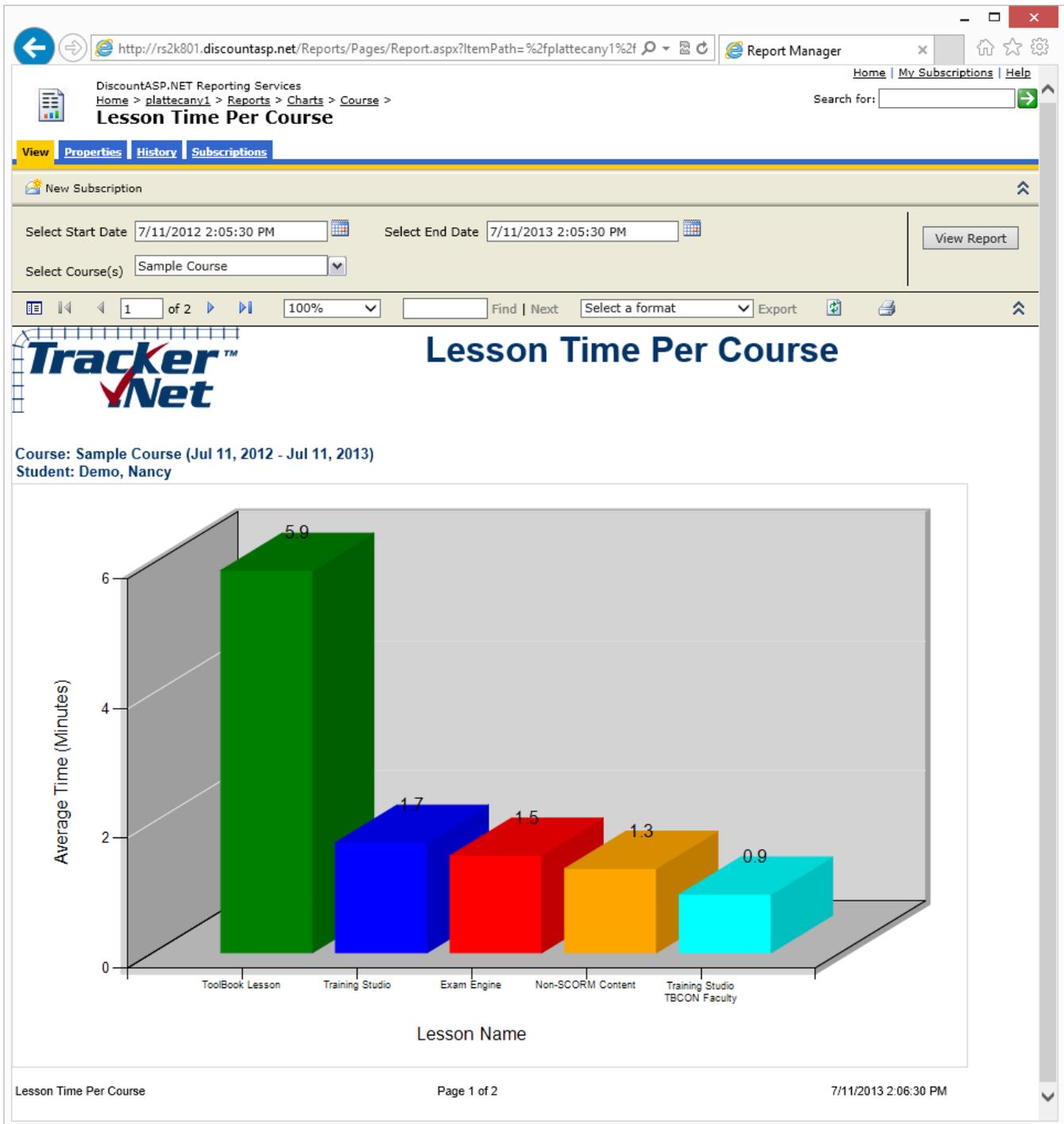
Tracker Reports Documentation

You then click on the report/chart that is of interest. In most cases, there are one or more *parameters* that you need to set for the report. For example, in the *Lesson Time Per Course* report shown below, you choose the start date⁷, end date, and course(s) you are interested in.

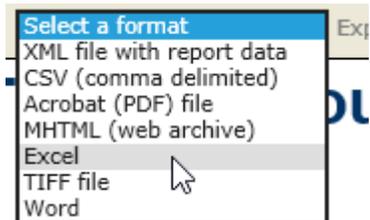
The screenshot shows a web-based form titled "New Subscription" with a navigation bar containing "View", "Properties", "History", and "Subscriptions" tabs. The form includes two date pickers: "Select Start Date" (7/11/2012 2:05:30 PM) and "Select End Date" (7/11/2013 2:05:30 PM). A "View Report" button is positioned to the right. Below the dates is a "Select Course(s)" dropdown menu, which is currently open and displays "Sample Course" with a checked checkbox and a mouse cursor hovering over it.

Once you have set the parameters, click the “View Report” button to actually display the report/chart. You can see an example below.

⁷ All the reports default to one year ago from the current date for the *start date* and the current date for the *end date*. You can change this in the *.rdl file for the report/chart if you desire.



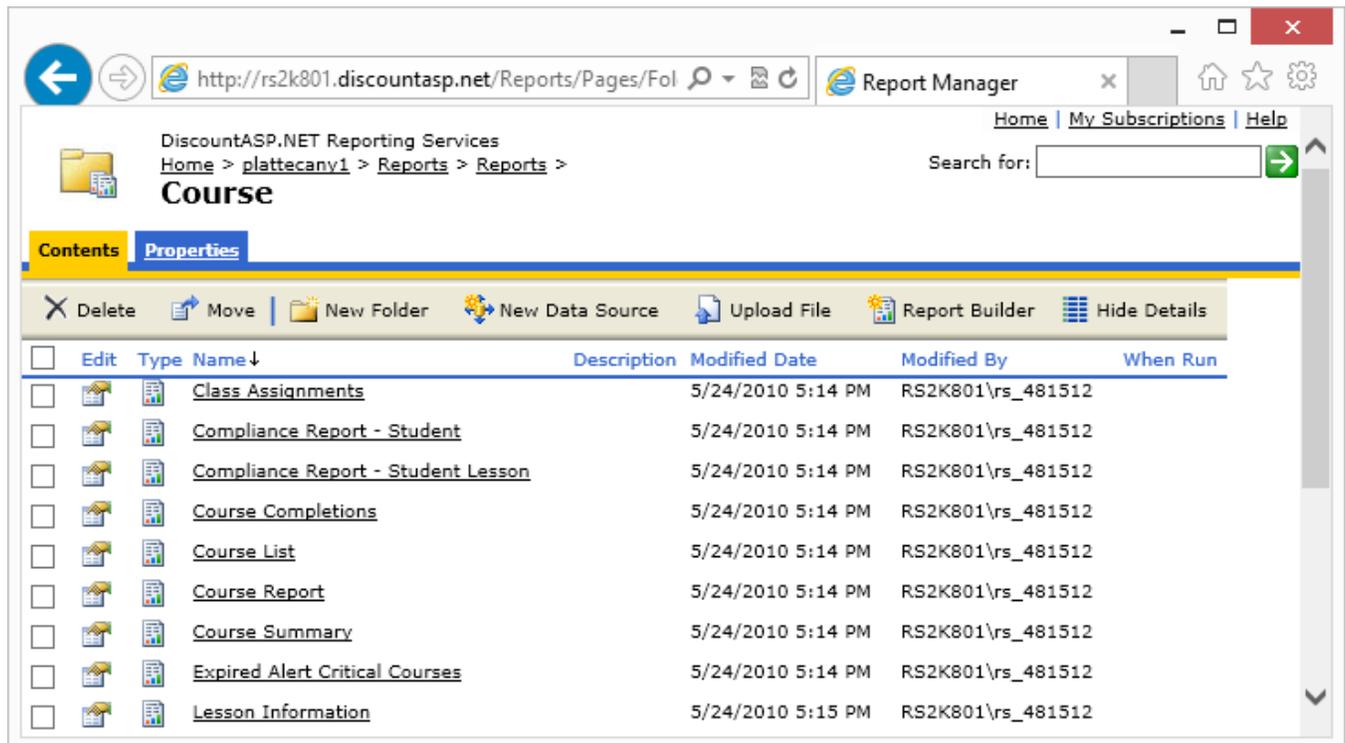
On the far left is the *Document Map*. It allows you to select from the items and jump to them in the report. You can show or hide the map with the far left button on the toolbar. Next to the *Document Map* button are navigation buttons for moving to the first, last, next, or previous page of the report. Next to those is the *Zoom* drop-down box. You can select



choices like 200%, Page Width, Whole Page, and more⁸. Still moving right on the toolbar, you can enter some text in the field and click the “Find” and “Next” links to jump to that text within the report. After that is the ability to export the report in various formats. These include XML, CSV, PDF, MHTML⁹ Excel, TIFF, and Word. Once you select your choice, click the *Export* link. Finishing up the toolbar are options to *Refresh* your data (rerunning the query on the database) and *Print* the current report/graph.

Setting up Subscriptions

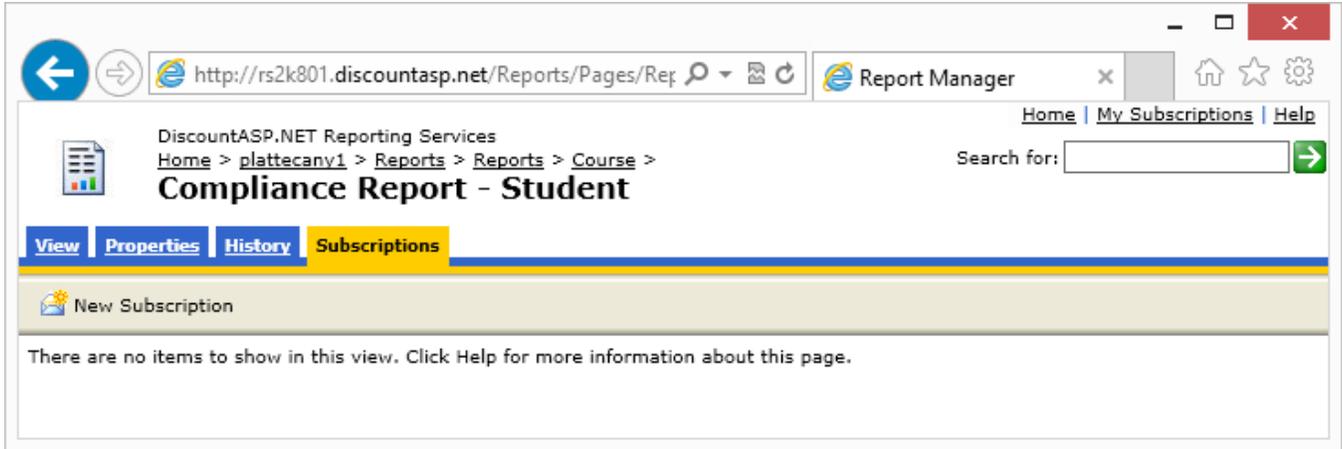
One of the nicest features of Reporting Services is the ability to email reports (this is not available for the version that comes with *SQL Server 2008 Express*). To use this feature, go to the properties for the report/chart for which you want to set up the description. To get to the properties, navigate to the desired folder and click the “Show Details” button on the toolbar. That will give a view like this:



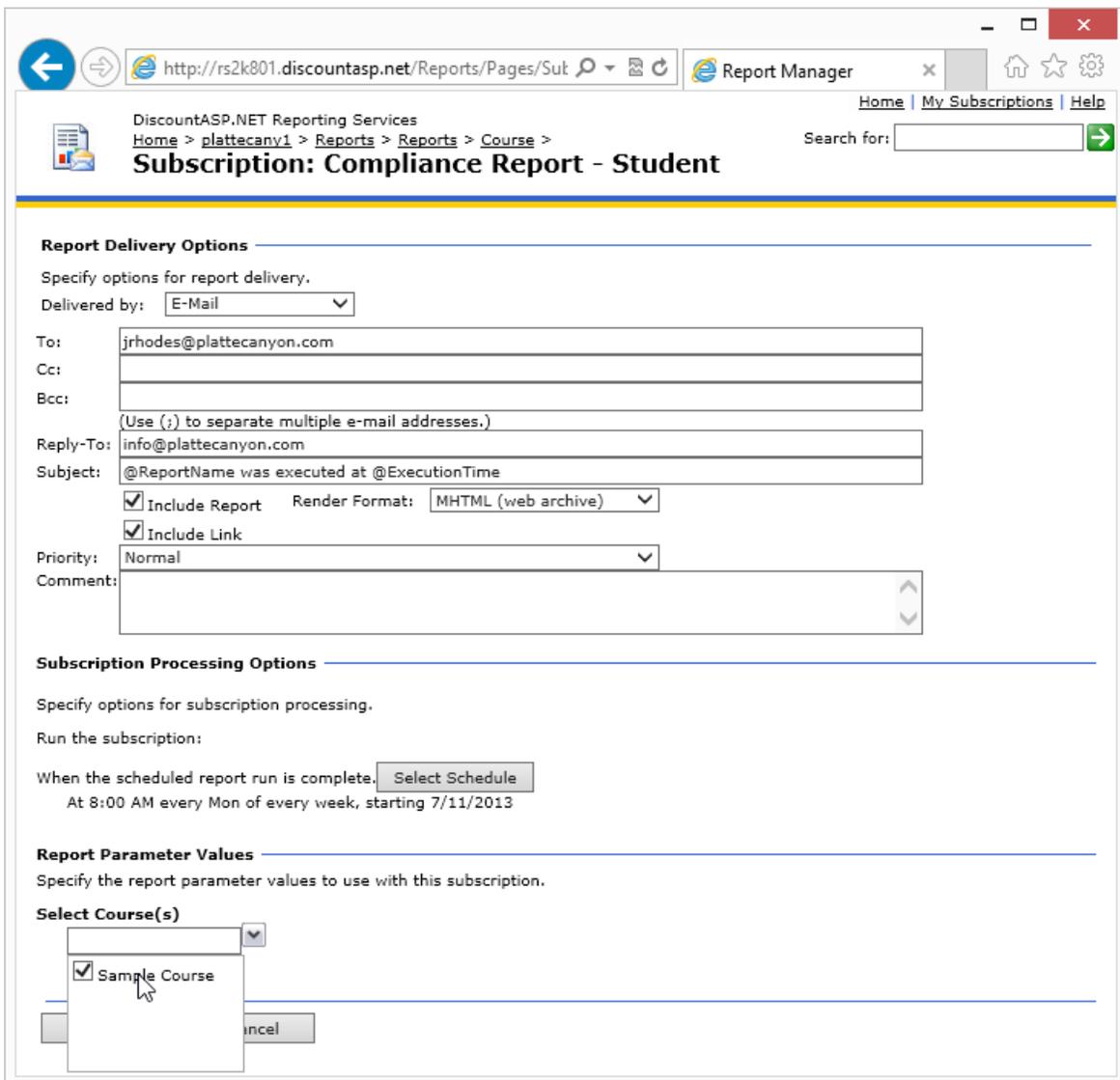
To set up the “Compliance Report – Student” report, we click the properties graphic underneath the Edit column. When then click the Subscriptions tab as shown below:

⁸ Those of you familiar with reports within Tracker.Net might notice that these use the same Microsoft *Report Viewer* control as some of the built-in reports. This should allow your users to be already familiar with the report functionality.

⁹ This is the default format that is emailed to users when you set up subscriptions, which we will cover shortly.



We then click the “New Subscription” button. That gives us the screen shown below. Most of the options are self-explanatory, but we will go over the rest.



Tracker Reports Documentation

The choices for delivery are E-mail and Windows File Share. The latter means that the report will be written to a shared folder on your network. If the user has a shortcut to that file, she could launch it each morning to get the most current copy. If you are running SharePoint, you can deliver to a SharePoint library as well.

The render formats are the same as for exporting reports. We recommend the default MHTML as then the report will show up nicely in the email itself. However, users may prefer Excel or PDF. Including a link means that the user can follow the link to view and interact with the report in the browser.

The schedule defaults to every Monday at 8 AM. But you can change it by clicking the “Select Schedule” button, giving the screen below.

DiscountASP.NET Reporting Services [Home](#) | [My Subscriptions](#) | [Help](#)
Home > [plattecanv1](#) > [Reports](#) > [Reports](#) > [Course](#) > Search for:

Subscription: Compliance Report - Student

Use this schedule to determine how often this report is delivered.

Schedule details

Choose whether to run the report on an hourly, daily, weekly, monthly, or one time basis.
All times are expressed in (GMT -07:00) Pacific Daylight Time.

Hour
 Day
 Week
 Month
 Once

Daily Schedule

On the following days:
 Sun Mon Tue Wed Thu Fri Sat

Every weekday

Repeat after this number of days:

Start time: : A.M. P.M.

Start and end dates

Specify the date to start and optionally end this schedule.

Begin running this schedule on:

Stop this schedule on:

Finally, you can set any of the parameters of the report. In our case, we can set which course(s) to include in the report. This feature can be particularly helpful if a user is interested in a particular

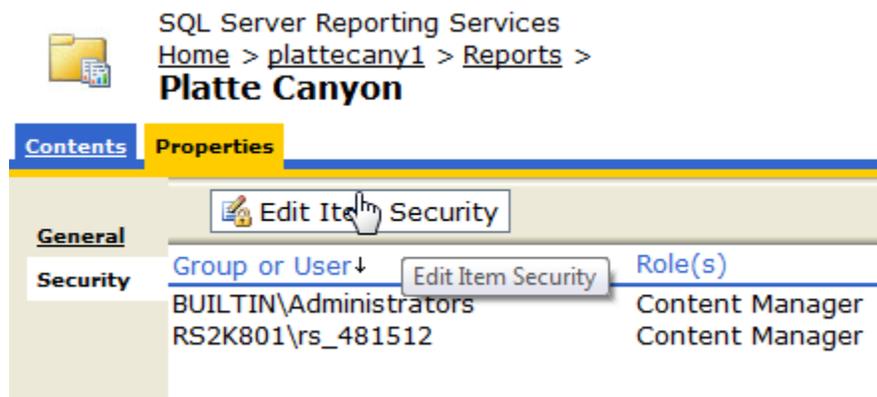
organization, class, course, lesson, or student. You can set that parameter and hide all other data. This is similar to the idea of a linked report, as we will discuss in the next section.

The ability to create subscriptions depends on your user rights within *Report Manager*. Assuming you have rights, you can also create “Data-Driven Subscriptions” to mass mail reports to a list of emails stored in a data source. This can be helpful if the desired list of recipients changes frequently.

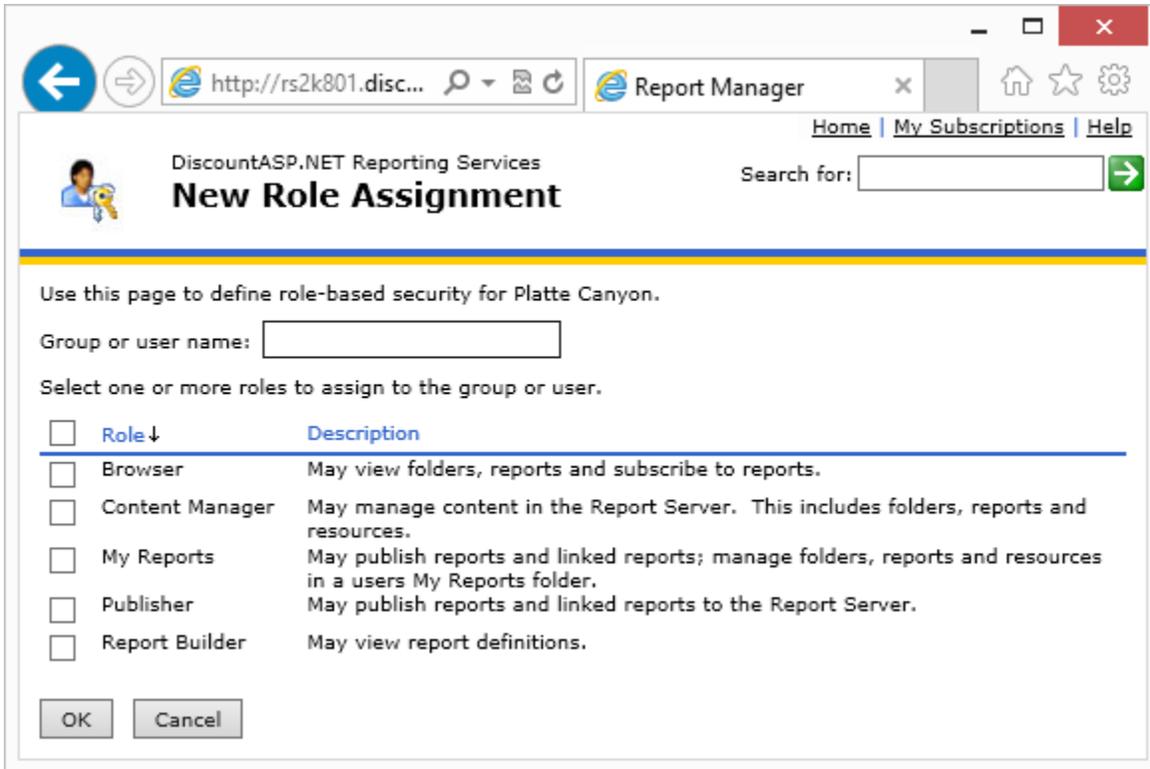
Creating Linked Reports

Linked reports involve the ability to customize the same report for different users. For *Tracker Reports*, the most common use of linked reports is to limit information to a particular organization. For example, one of the organizations in our database is Platte Canyon Multimedia Software Corporation. Suppose that I want to create a “Platte Canyon” folder that only specific user(s) can access. In that folder would be organization reports limited to only Platte Canyon. The user(s) would NOT have access then to the *Reports* or *Charts* directories, since those have reports with data from other organizations. Let’s go through these steps in turn.

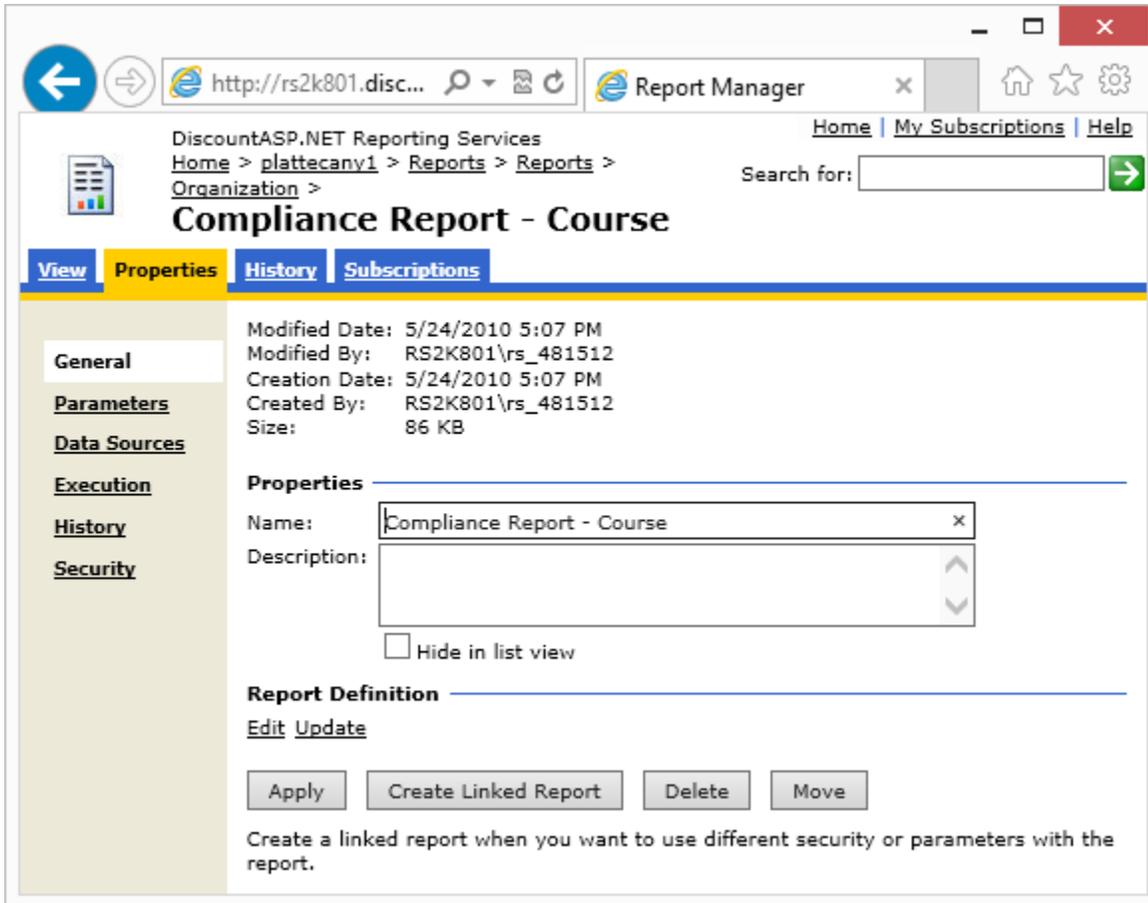
1. We create a new folder called “Platte Canyon.” We go to its properties and then to the Security link. Then click the “Edit Item Security” button as shown to the right.



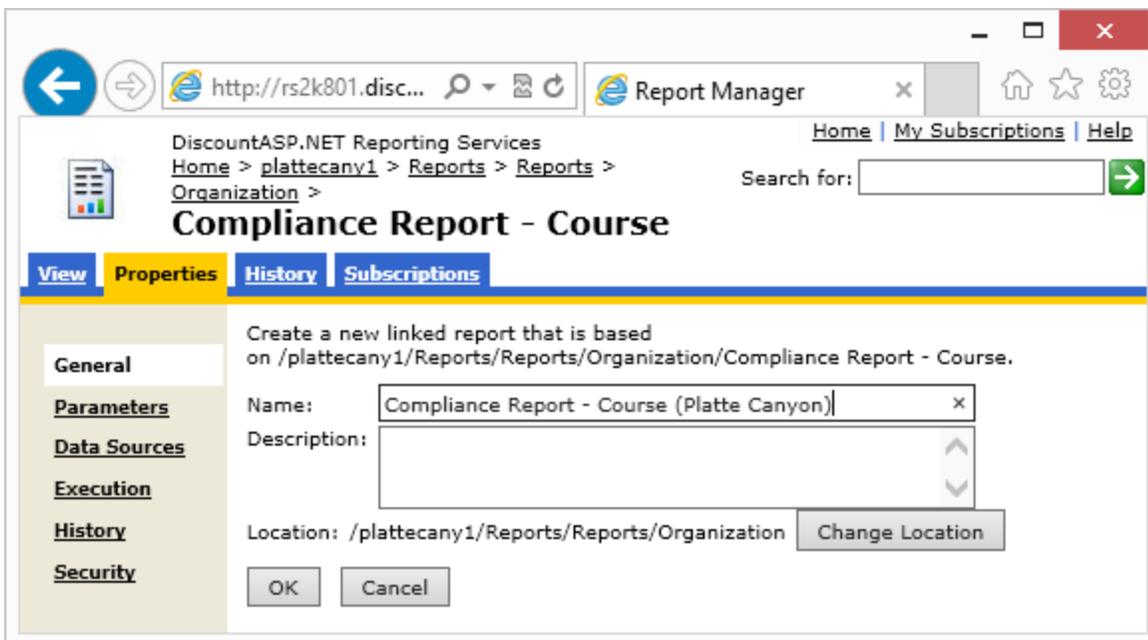
2. You can edit the permissions for any of the users listed or click the “New Role Assignment” button on the toolbar. This shows the screen below.



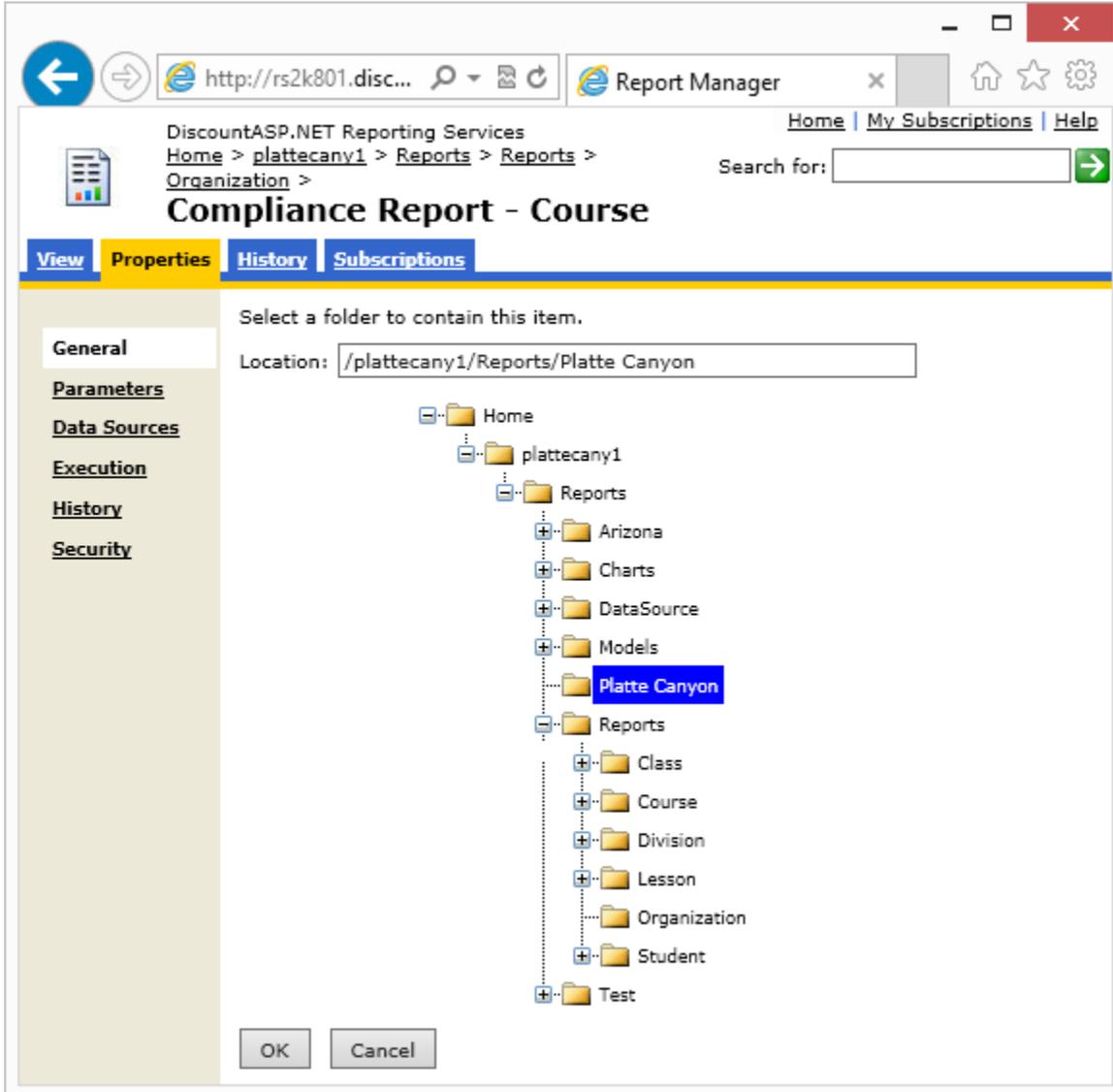
3. You can give the new group or user “Browser” or even more rights to this folder.
4. We will now link the “Compliance Report – Course” report to the Platte Canyon folder. To do that, we go to the report properties and click the “Create Linked Report” button as shown below.



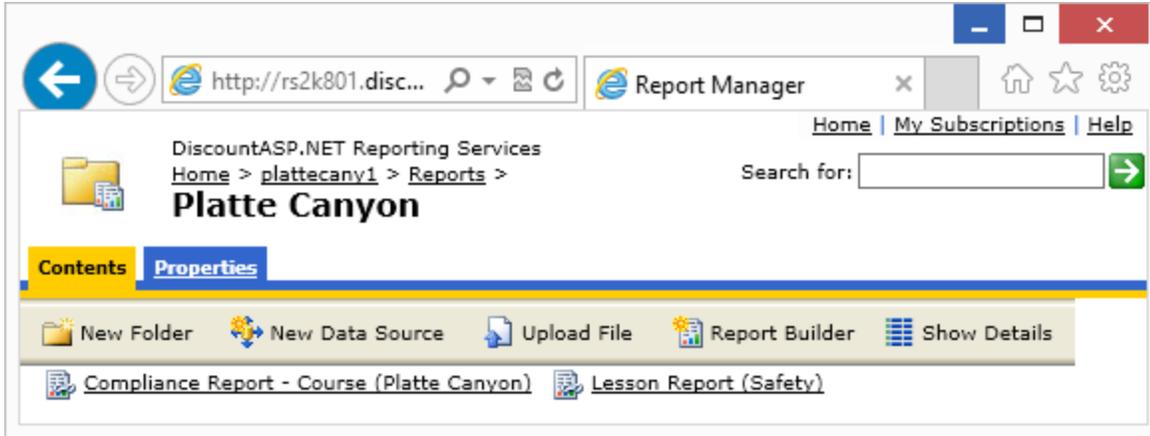
5. We next see the screen shown below.



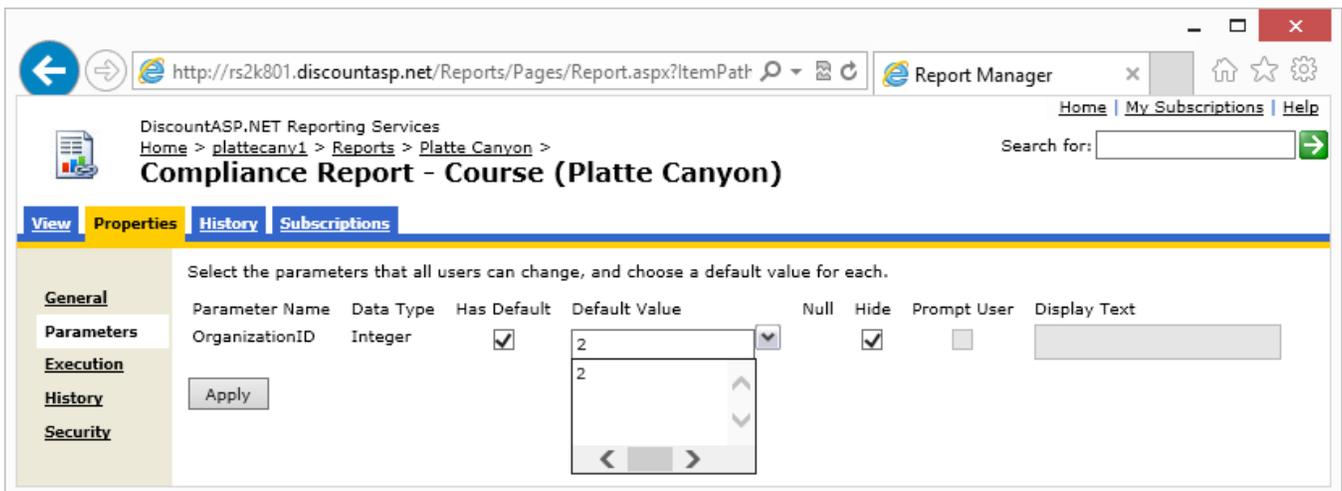
6. We enter a name and then click the “Change Location” button.



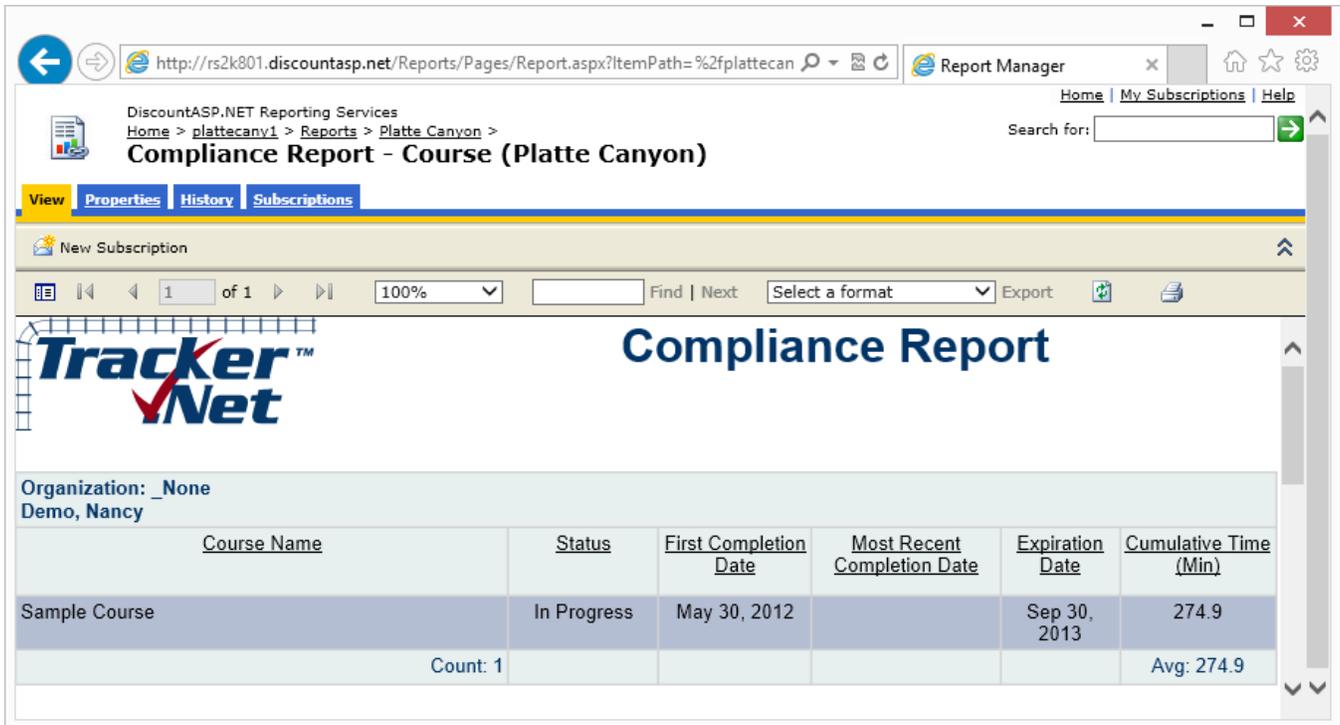
7. We select “Platte Canyon” from the list. We then click “OK.”
8. To set the parameters, we need to navigate back to the Platte Canyon directory. The icon next to the new report means that it is linked report.



- Click the “Show Details” button, click Properties, and then click on the Parameters link. You will see a screen like the one below. This part is a bit tricky. Check the “Hide” box and the “Has Default” box. Then click the “Default Value” drop-down box. We would expect to get a list of available organizations, but unfortunately the list is blank. To proceed, you’ll need to go to the original database and find the correct numeric identifier for what we want. In our case, we go to the *Tracker_Organizations* table and find that the correct *OrganizationID* is 2. We type in that value and click the drop-down box to get it to “take.” We then click the “Apply” button.



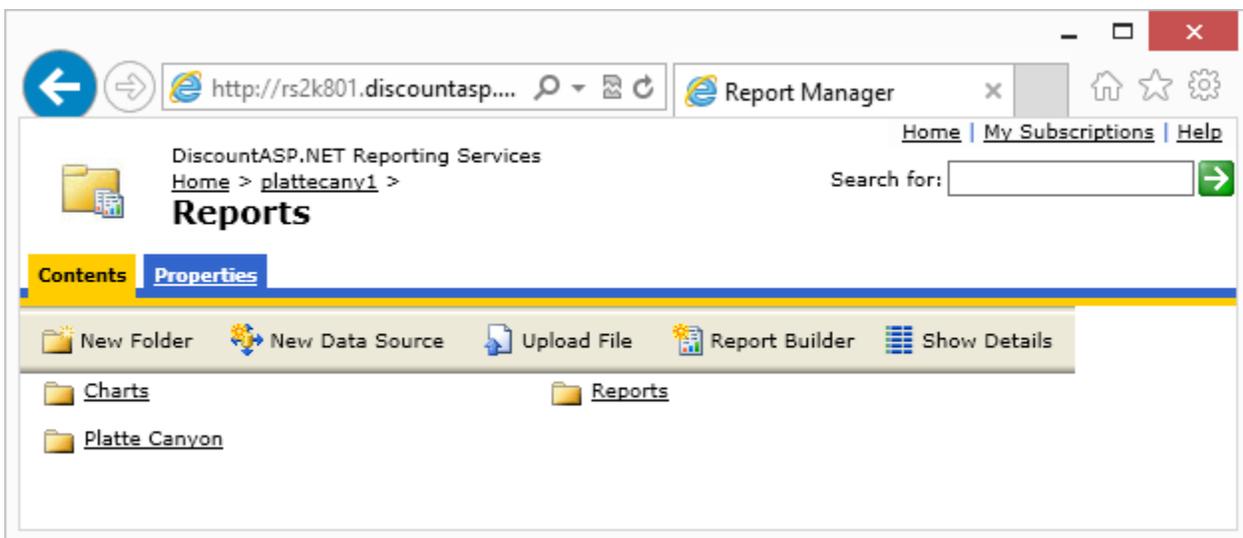
- We click the “View” tab to see the results. Notice how there is no “Select Organization(s)” drop-down.



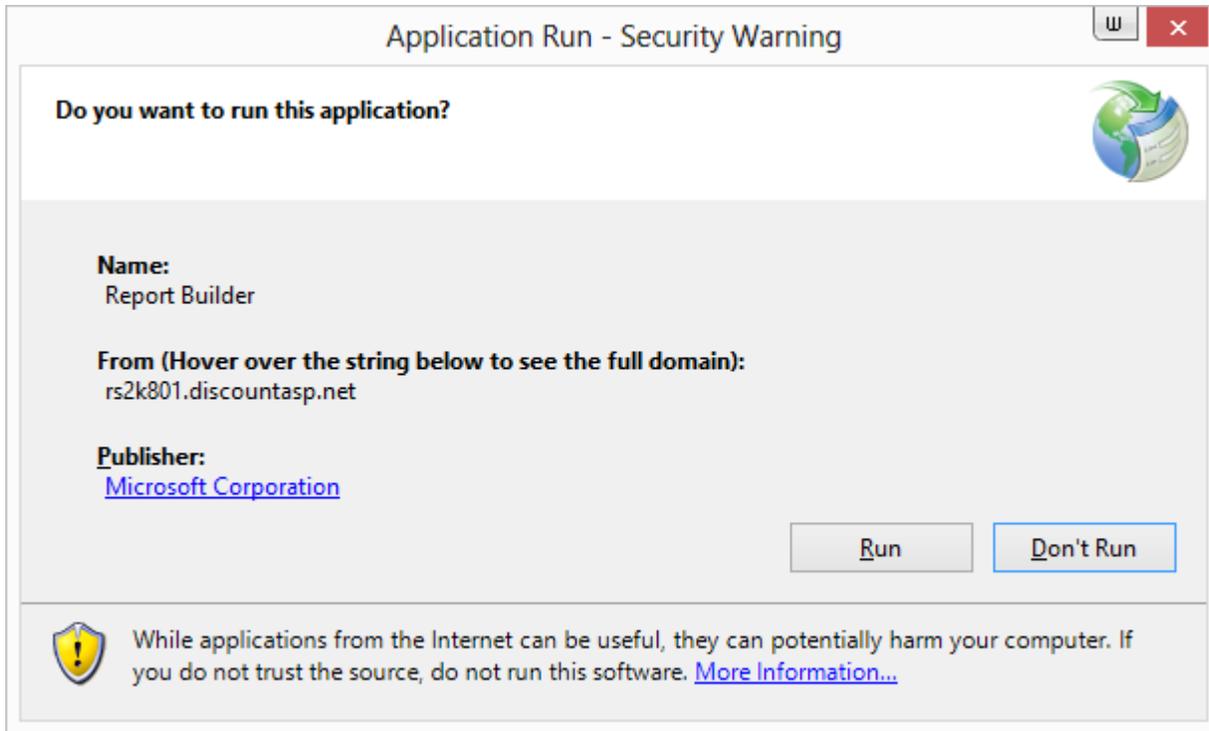
We could link additional reports/charts to this directory using the same process.

Ad-Hoc Reporting and Report Builder

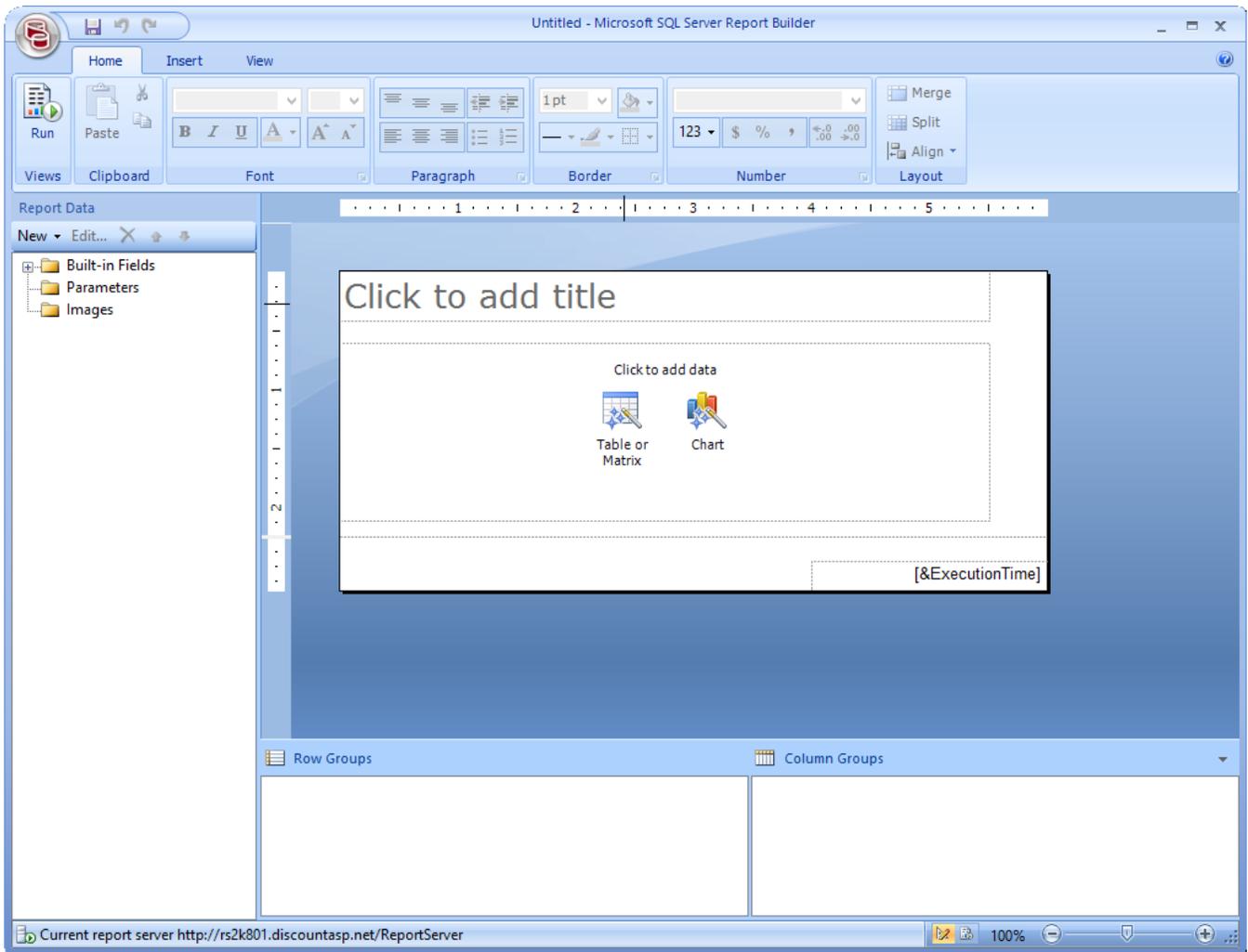
Our last topic is ad-hoc reporting. This is again not available in *SQL Server 2008 Express*. Ad-hoc reporting is for advanced users. We previously showed you how to upload the *Tracker Model* to *Report Manager*. Once you have done that, you can click on the *Report Builder* button on the toolbar:



The *Report Builder* application is a "Click Once" installation of a .NET executable. You may be prompted to authorize it and will see a download screen like the one below.

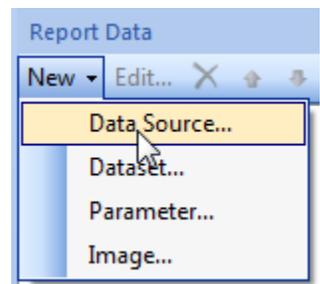


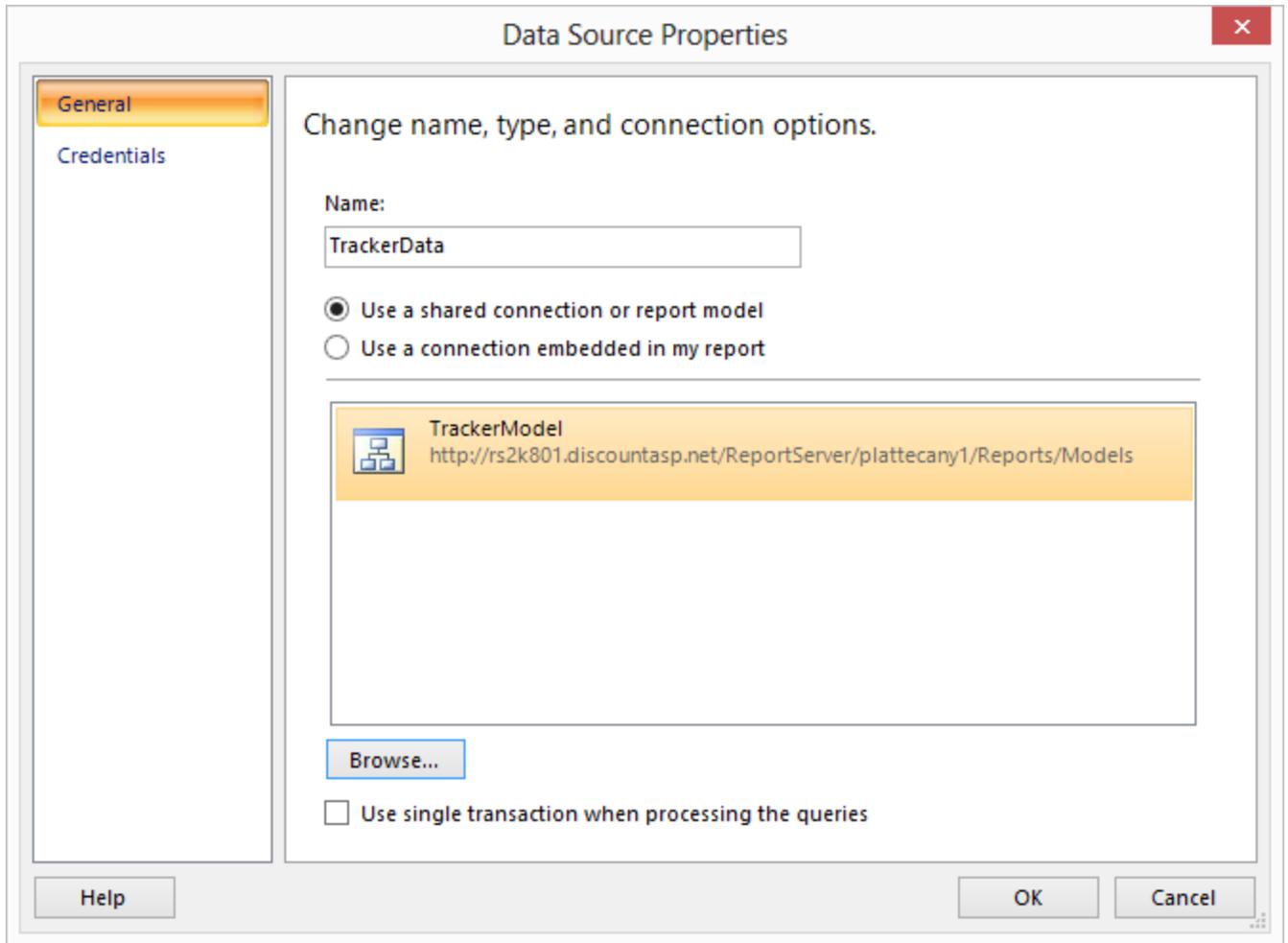
Once it is finished, you see an application similar to the report tools in Microsoft Access and a bit like a simplified version of Visual Studio. You can also install the Report Builder via a free download from Microsoft.



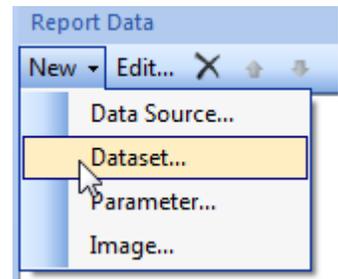
Let's create a simple report to see how to proceed.

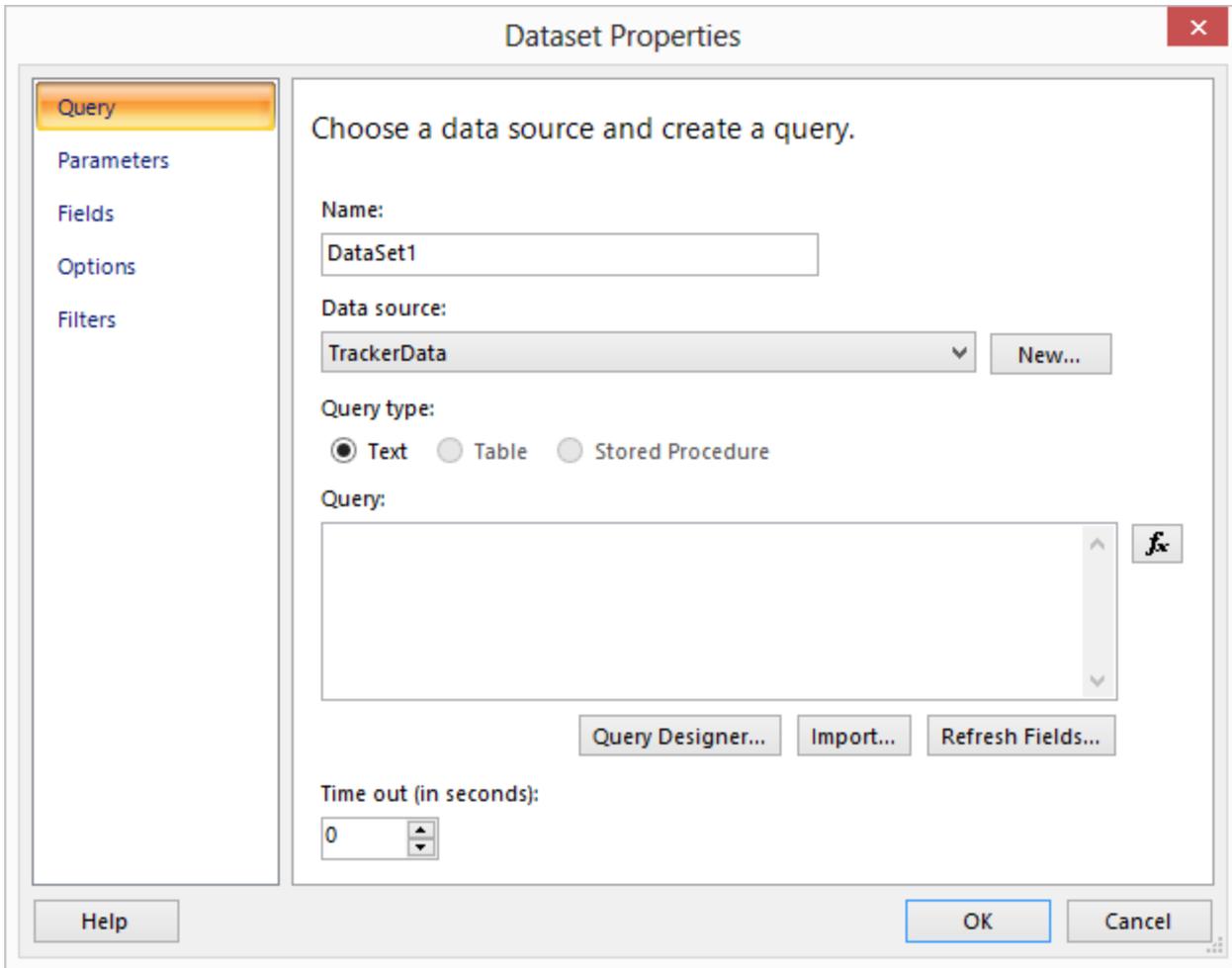
1. Click on the *New* button in the Report Data window and select "Data Source" as shown to the right.
2. If *TrackerModel* is not shown as a connection, click on the "Browse..." button and navigate to the location of it. For example, it might be in */Reports/Models*. You can name the data source as well.





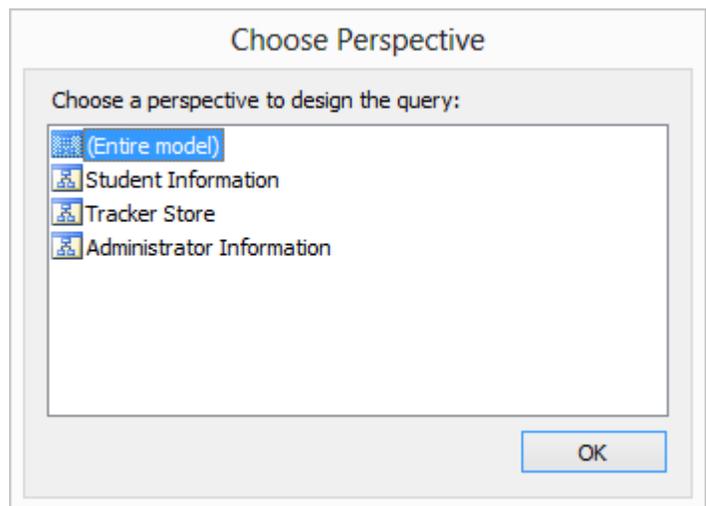
3. Next, follow a similar process to create a new Data Set, which is a .NET representation of one or more database tables. It will already default to the data source that we created in the previous step.





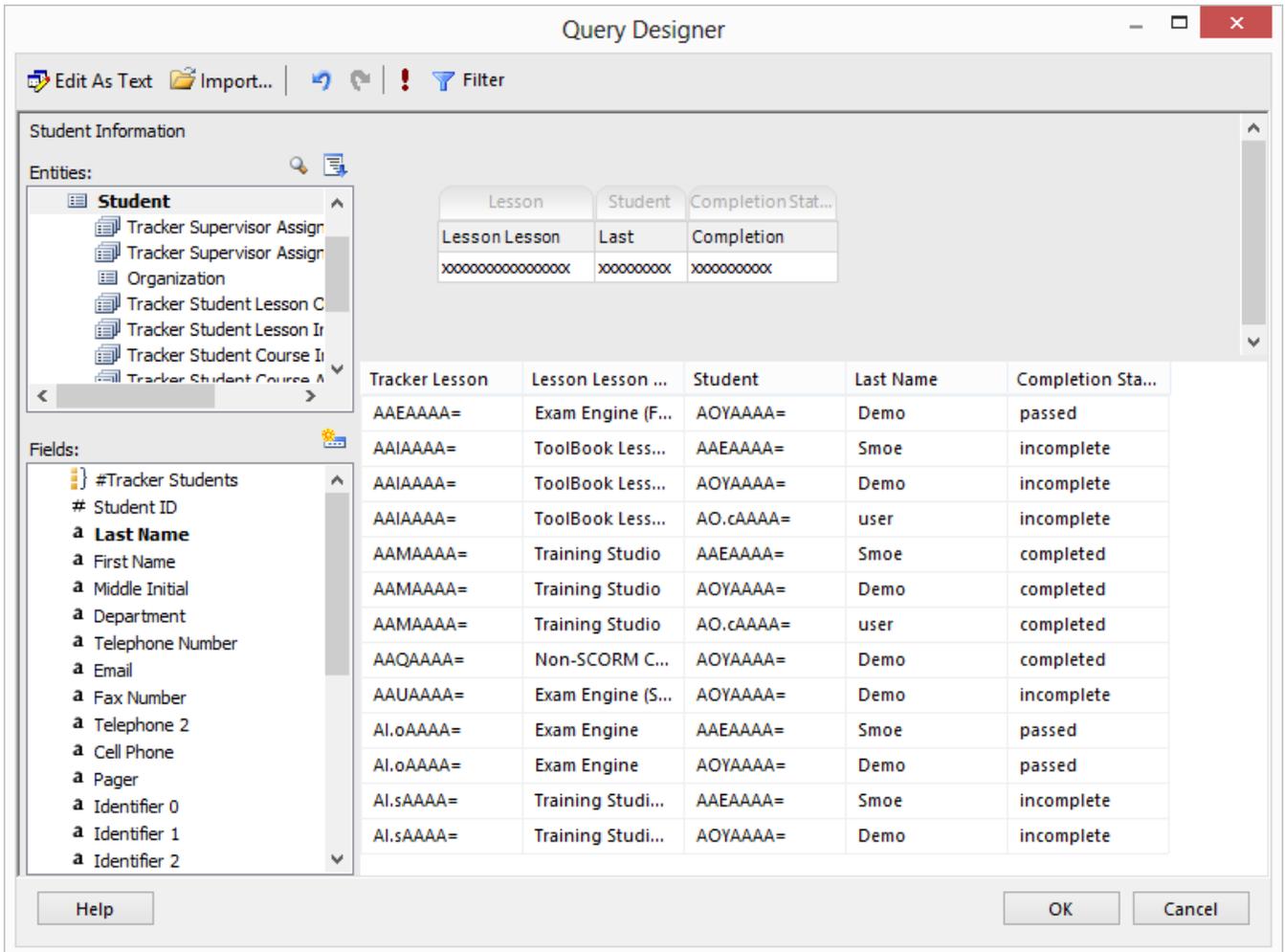
4. We click on the “Query Designer...” button to build the query that we need for our report.

5. This is where the idea of the model comes into focus. The *TrackerReportsModel* solution has information on the various interrelated tables in the database. In addition, it has *Student Information*, *Tracker Store*, and *Administrator Information* perspectives as shown below. These perspectives organize just the information a report writer might need. For example, *Student Information* excludes tables like *Tracker_Globals* and *Tracker_LanguageSettings*.

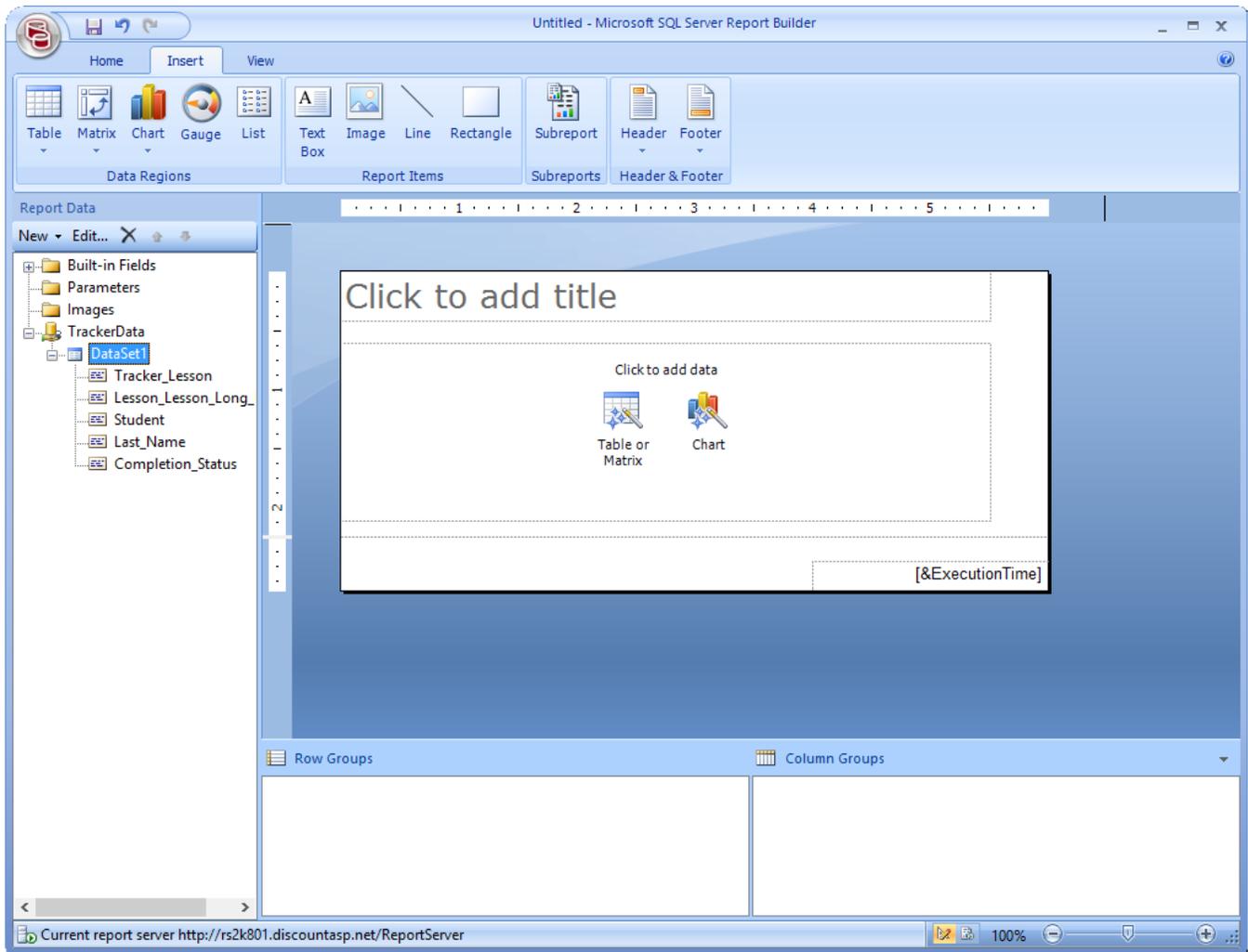


6. Once we have picked our desired perspective, we see a list of Entities (basically tables) and Fields (columns as well as some

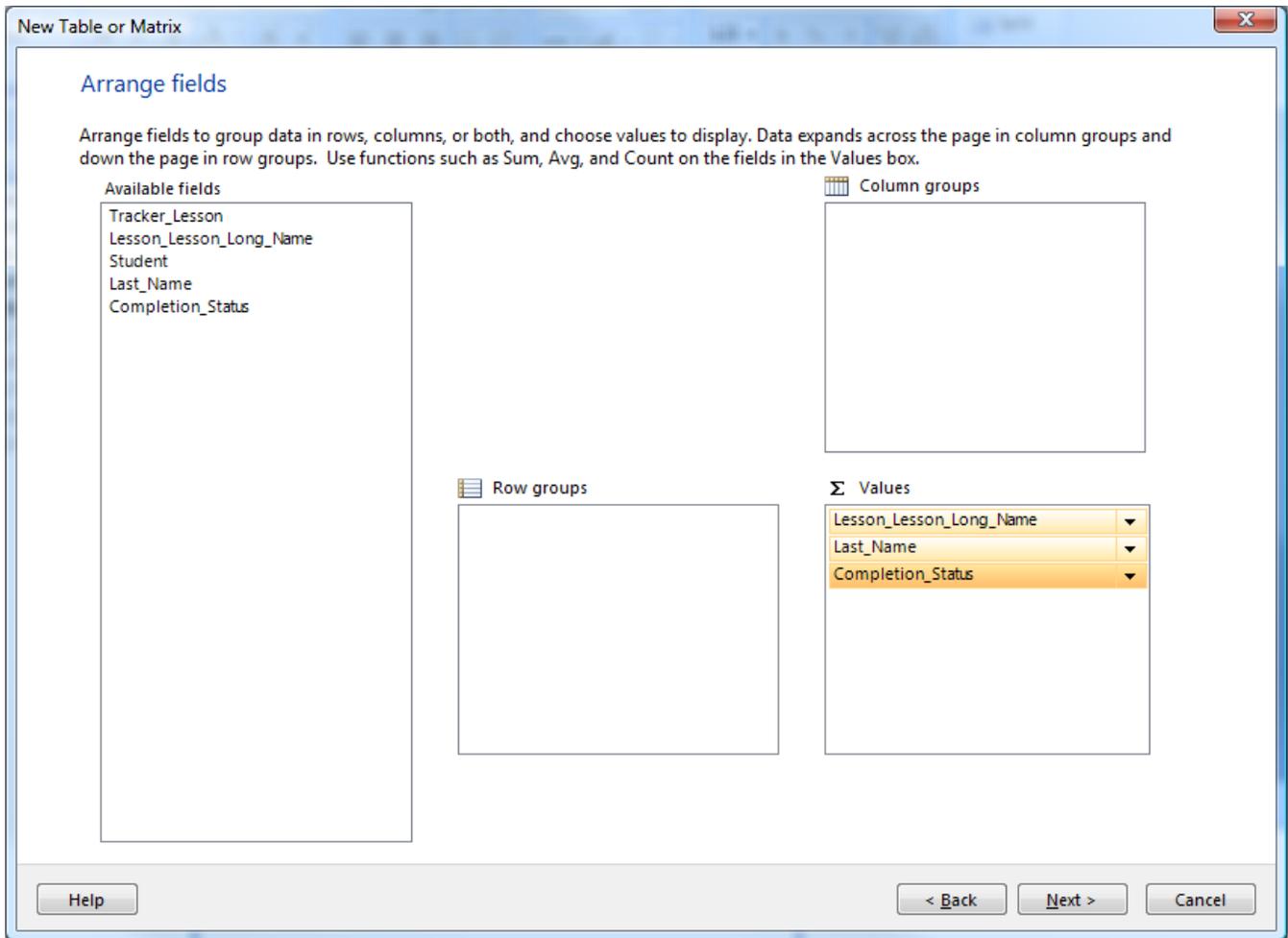
aggregates like Count and Average). When you click on an Entity, then related entities are shown automatically. For example, in the screen capture below, we started with *Tracker Lessons*, clicked on *Tracker Student Lesson Information*, and then on *Tracker Students* to get the data we want. Notice that the weird AAEEAA and related columns are for use by the model and not something we want to display in our report.



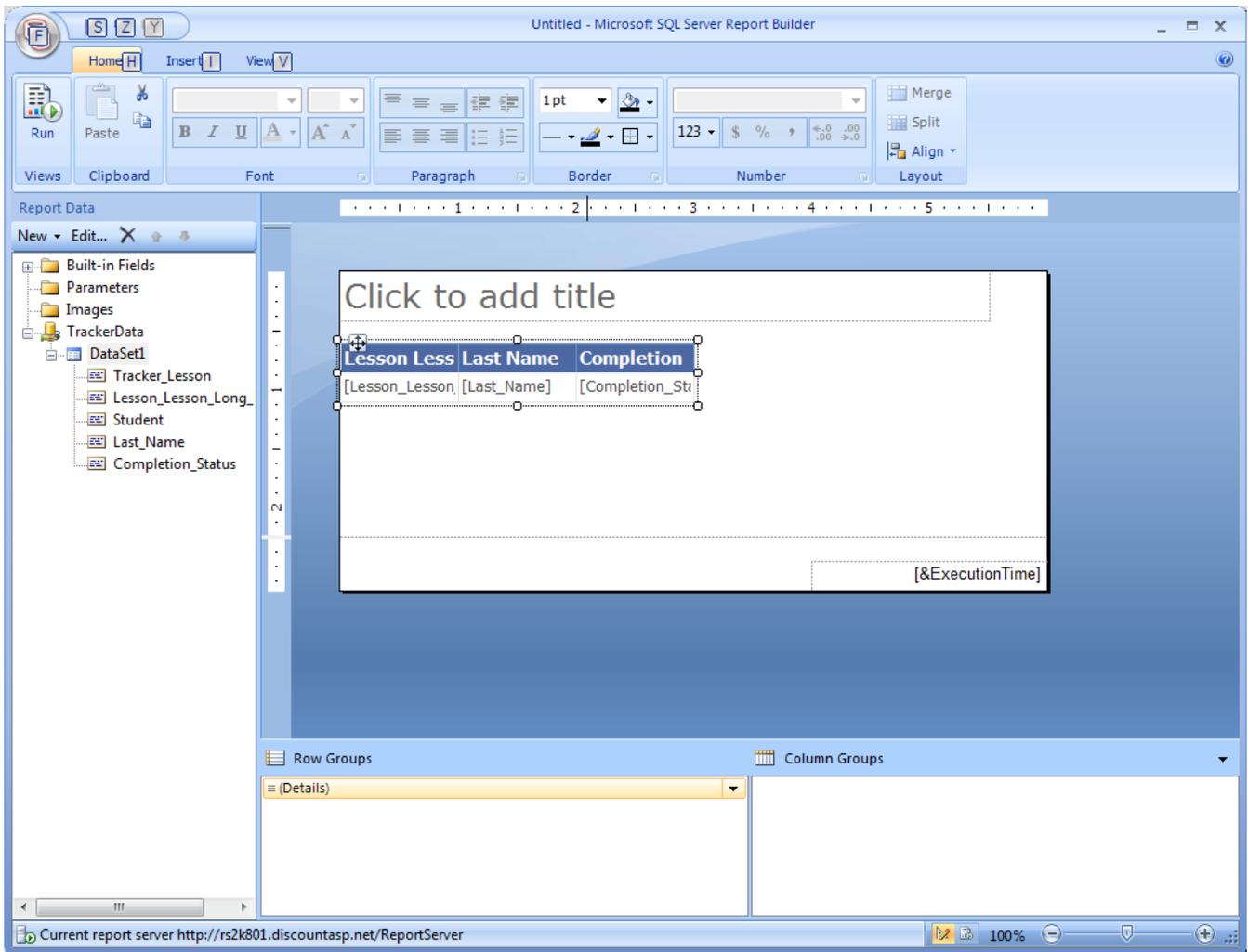
- Once we click “OK” on both dialog boxes, we are back to the main screen but now with data available.



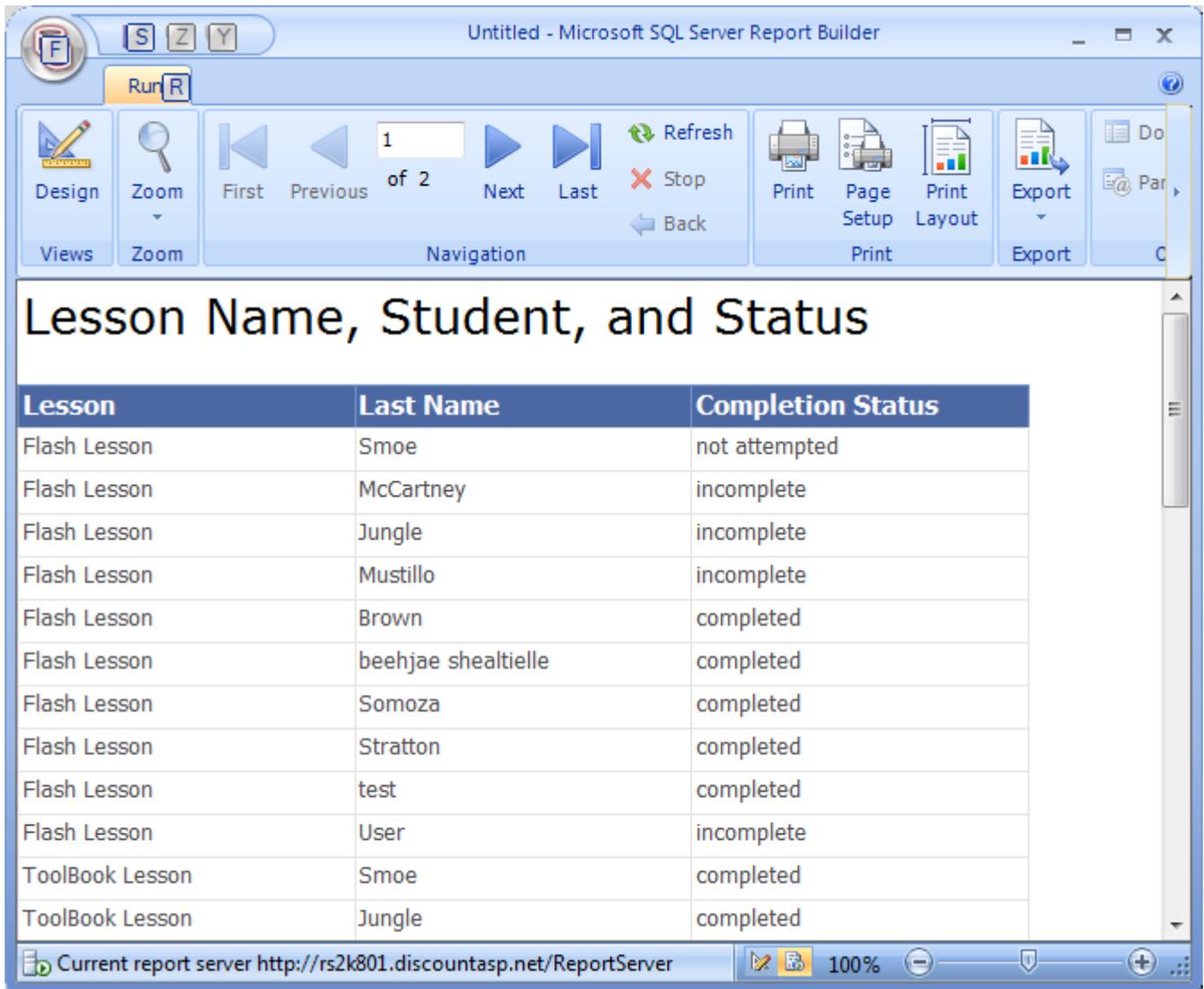
8. We next click on either “Table or Matrix” or “Chart.” In our case, we’ll do the former. *Report Builder* gives us a wizard. To use it effectively will take some practice and perhaps some outside references. Here, we will build a simple table with only values and no groups.



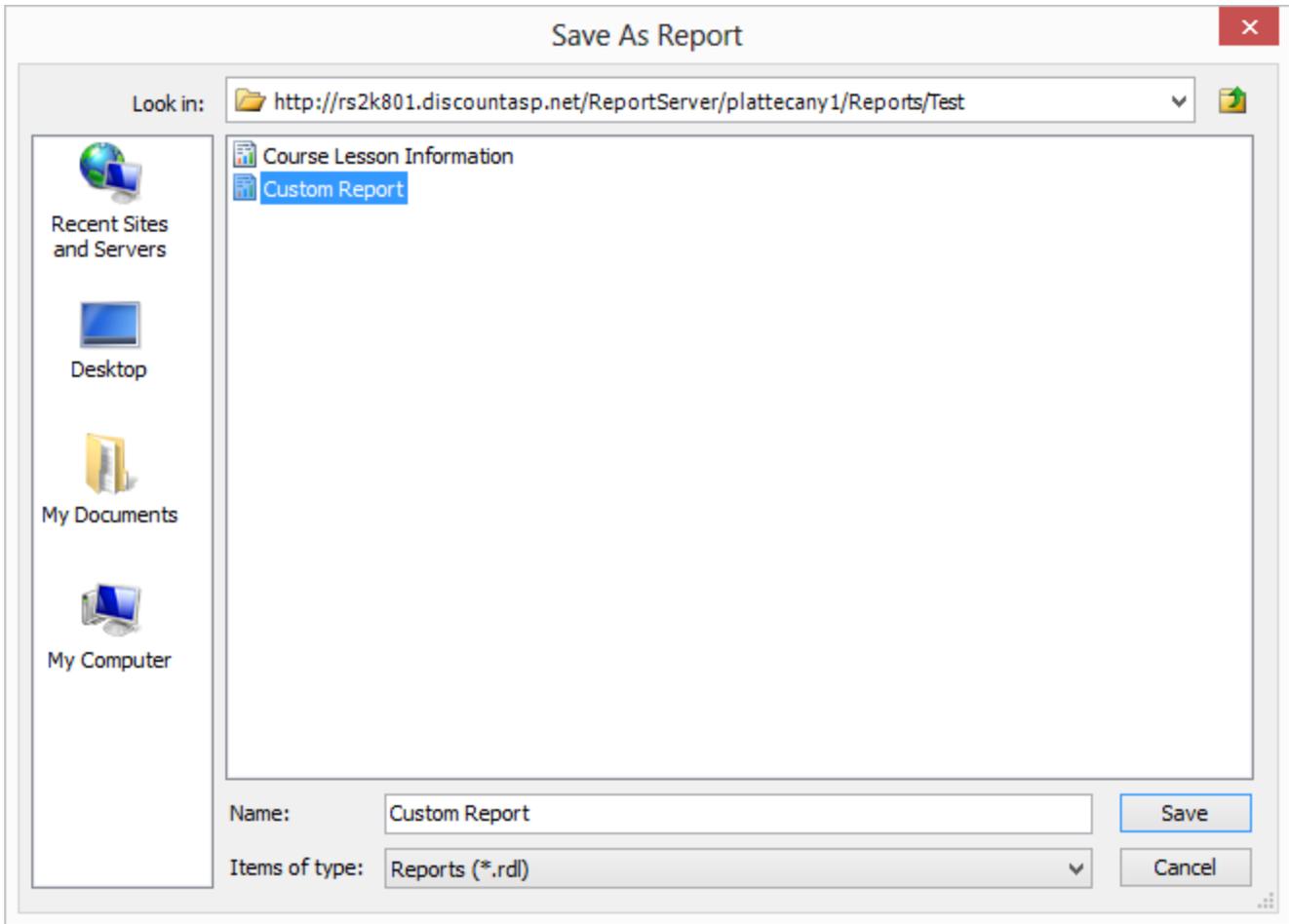
9. Taking the defaults for everything else, we get back to the design surface.



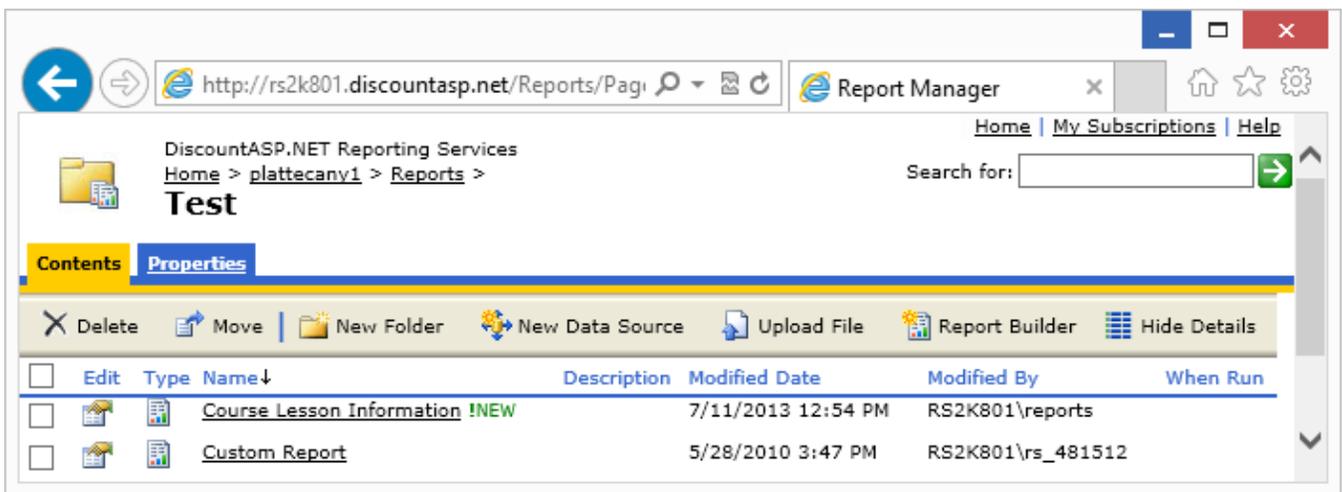
10. With a little formatting and adding a title, we get this result (after clicking the “Run” button).



11. We can then save the report on either the server or our local machine. In the screen capture below, we save it to the *Test* directory we created earlier in this document.



12. Any users with rights to that folder can now run the custom report. Here is how it looks in *Report Manager*.



13. Finally, here is our ad-hoc report from within *Report Manager*.

The screenshot shows a web browser window titled "Report Manager - Internet Explorer provided by Dell". The address bar contains the URL "http://rs2k801.discountasp.net/Reports/Pages/Report.aspx?ItemPath=%2fpl". The page content includes a breadcrumb trail: "Home > plattecan1 > Reports > Test > Custom Report". There are navigation tabs for "View", "Properties", "History", and "Subscriptions". Below the tabs is a "New Subscription" section. A navigation bar shows "1 of 2" pages, a "100%" zoom level, and "Find | Next" options. The main content area features a table with the following data:

Lesson	Last Name	Completion Status
Flash Lesson	Smoe	not attempted
Flash Lesson	McCartney	incomplete
Flash Lesson	Jungle	incomplete
Flash Lesson	Mustillo	incomplete
Flash Lesson	Brown	completed
Flash Lesson	beehjae shealtielle	completed