

This article was brought to you by [4GuysFromRolla.com](http://www.4GuysFromRolla.com)! When you think ASP, think [4GuysFromRolla.com](http://www.4GuysFromRolla.com)!

<http://www.4GuysFromRolla.com>

To read the article online visit <http://aspnet.4GuysFromRolla.com/articles/031605-1.aspx>

An Introduction to Microsoft SQL Server 2000 Reporting Services

By *Don Franke*

Introduction

As programmers, when building data-driven Web sites we typically focus our energies on defining what information needs to be accepted by the end user, and ensuring that this data is properly entered into some backend data store properly via a Web interface. After all, if a Web application cannot correctly accept the data it is designed to collect, the usefulness of the site is greatly reduced. However, the usability of a data-driven site is not limited strictly to collecting the necessary information. Equally as important is analyzing and displaying the information in an easy-to-read manner, a task that is commonly referred to as *reporting*.

There are a number of third-party reporting products available, the most (in)famous one being [Crystal Reports](#). Recently, however, Microsoft released their own reporting solution software, [Reporting Services](#). Reporting Services is an add-on for SQL Server 2000, and has been made available by Microsoft since early 2004. Furthermore, Reporting Services is licensed as part of SQL Server 2000, which means if your company already has a licensed copy of SQL Server, you're in luck - Reporting Services can be installed and run on the same server for no additional license fee. More information on obtaining Reporting Services is available from Microsoft's site:

- [How to Obtain Reporting Services](#)
- [Reporting Services Downloads](#)

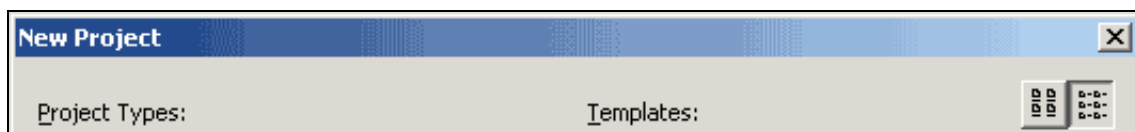
Also be sure to check out John Hancock's MSDN article, [Deliver User-Friendly Reports from Your Application with SQL Server Reporting Services](#).

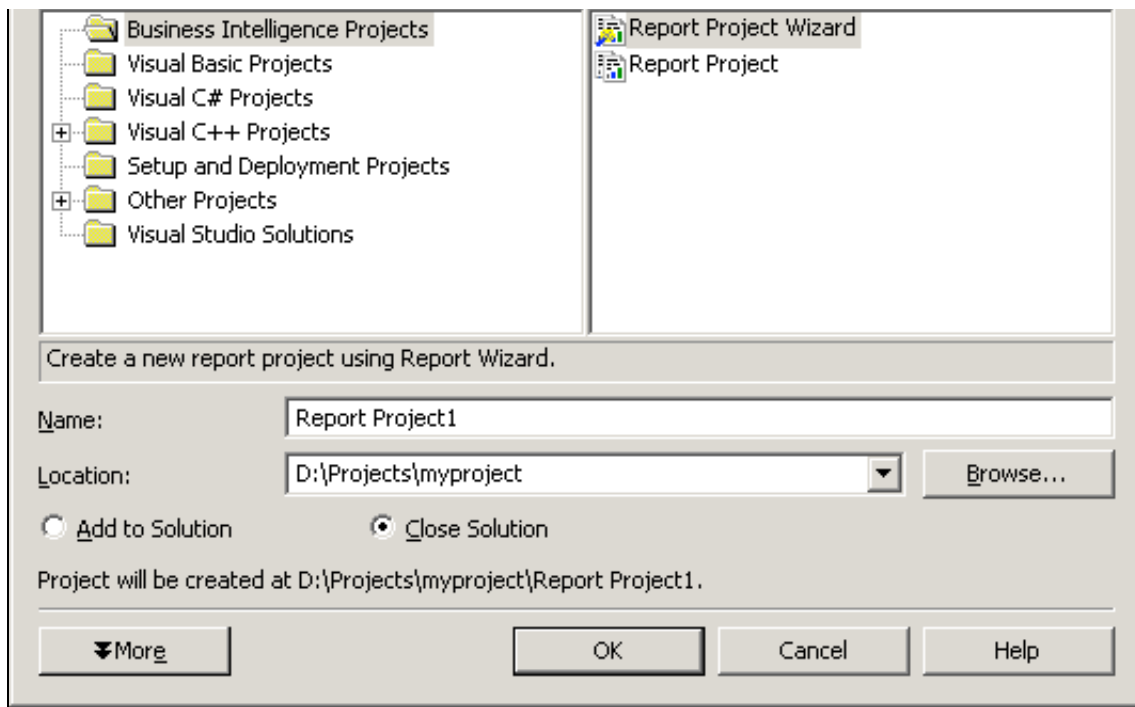
This article provides a simple walk-through on creating and displaying a report in an ASP.NET Web page using Reporting Services. It assumes that you have already downloaded and installed Reporting Services and that you are using Visual Studio .NET 2003 as your IDE. Read on to learn more!

Creating a Report with Reporting Services

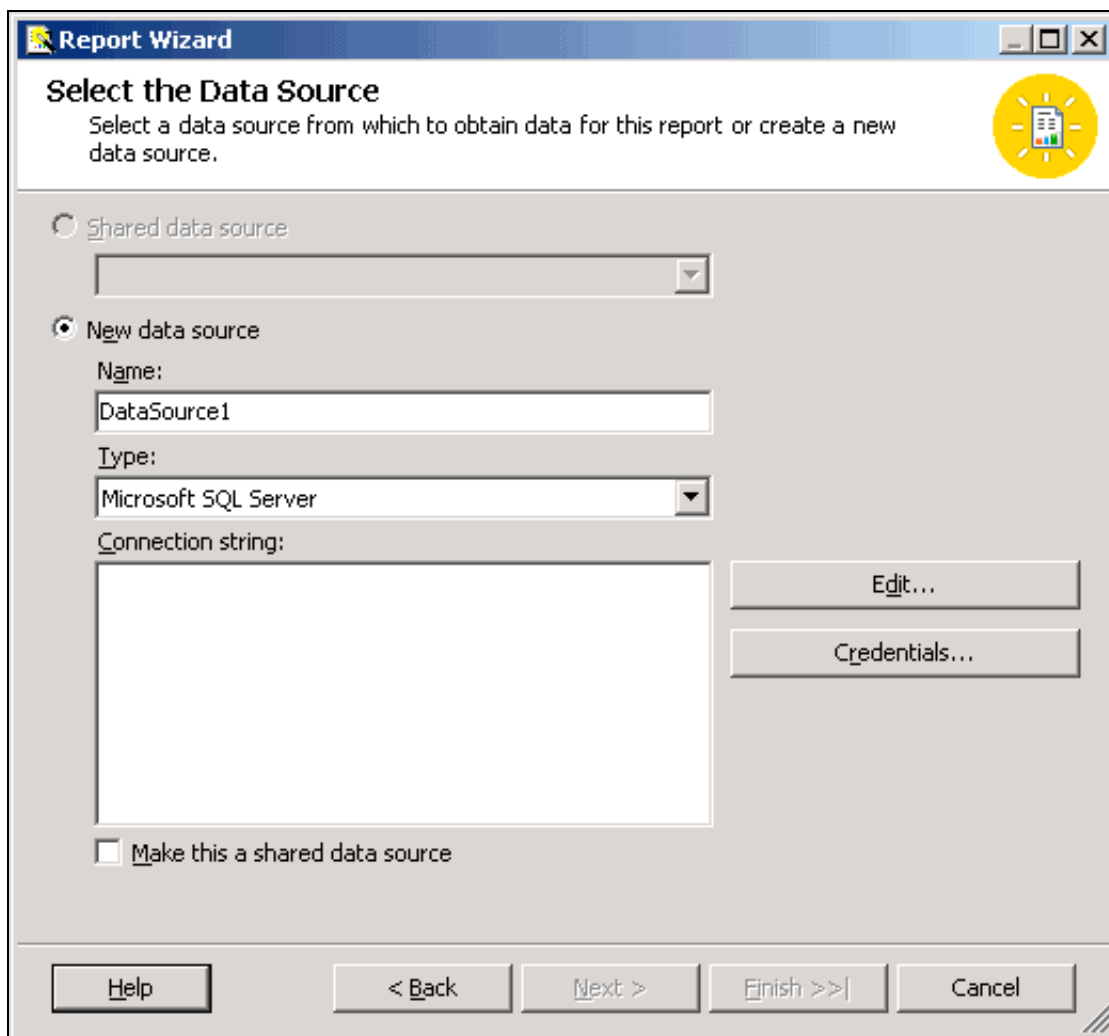
In order to use Reporting Services you must have both SQL Server 2000 and Visual Studio .NET 2003. Over the remainder of this article we'll be stepping through creating a report using the Northwind database installed with SQL Server as the store from which the report will pull its data. Once you have installed Reporting Services, the first step to creating a new report is to fire up Visual Studio .NET and create a new project.

For the Project Type select Business Intelligence Projects and for Templates select Report Project Wizard. Check your project name and location fields, then click OK. Note: when designing reports you are saving them locally. It is when your report is deployed that it is moved to the Report Server and made available to everyone else.



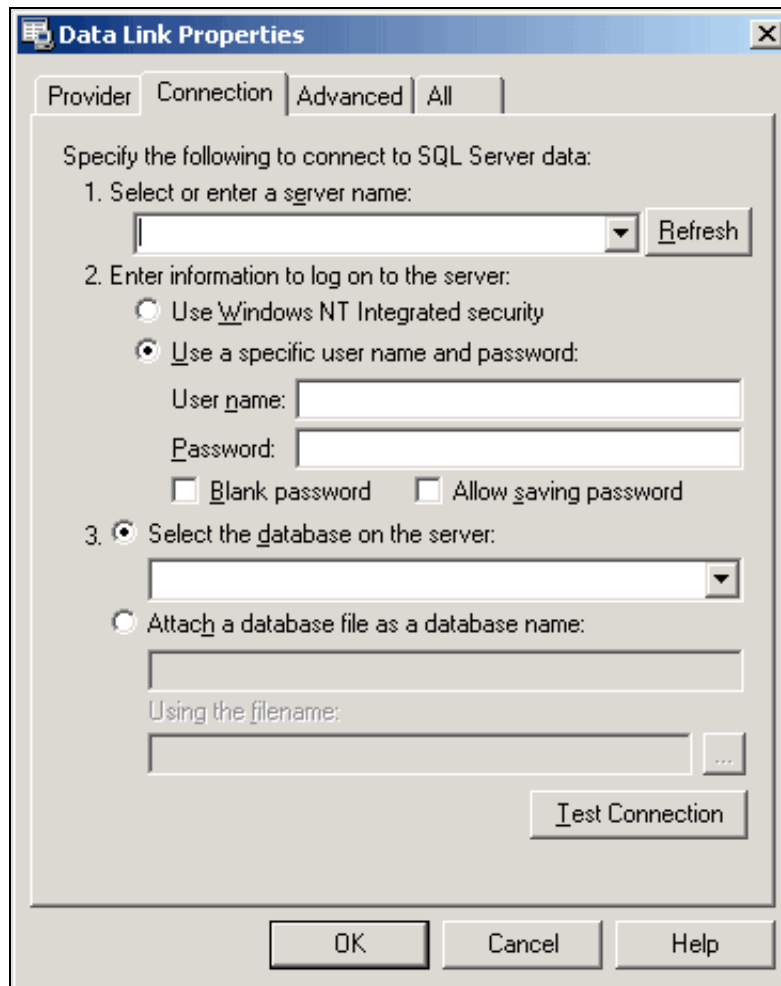


First we have to create a data source for the report. Since this is a new project you will have to create a new data source from scratch. Subsequent reports for this project can reuse this data source. Click Edit... (above the Credentials... button) to show the Data Link Properties screen.

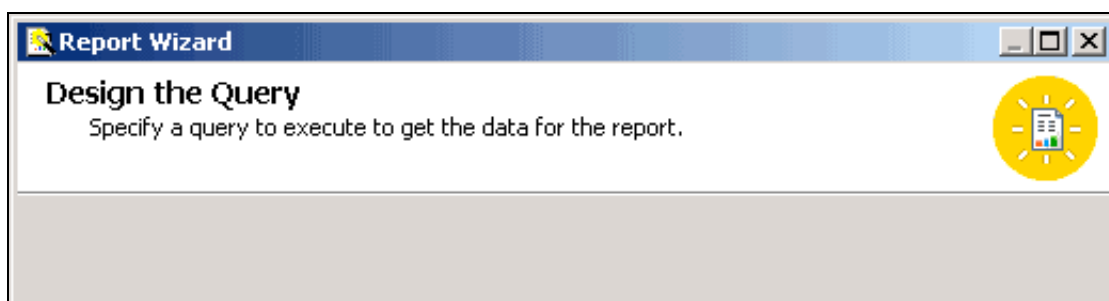


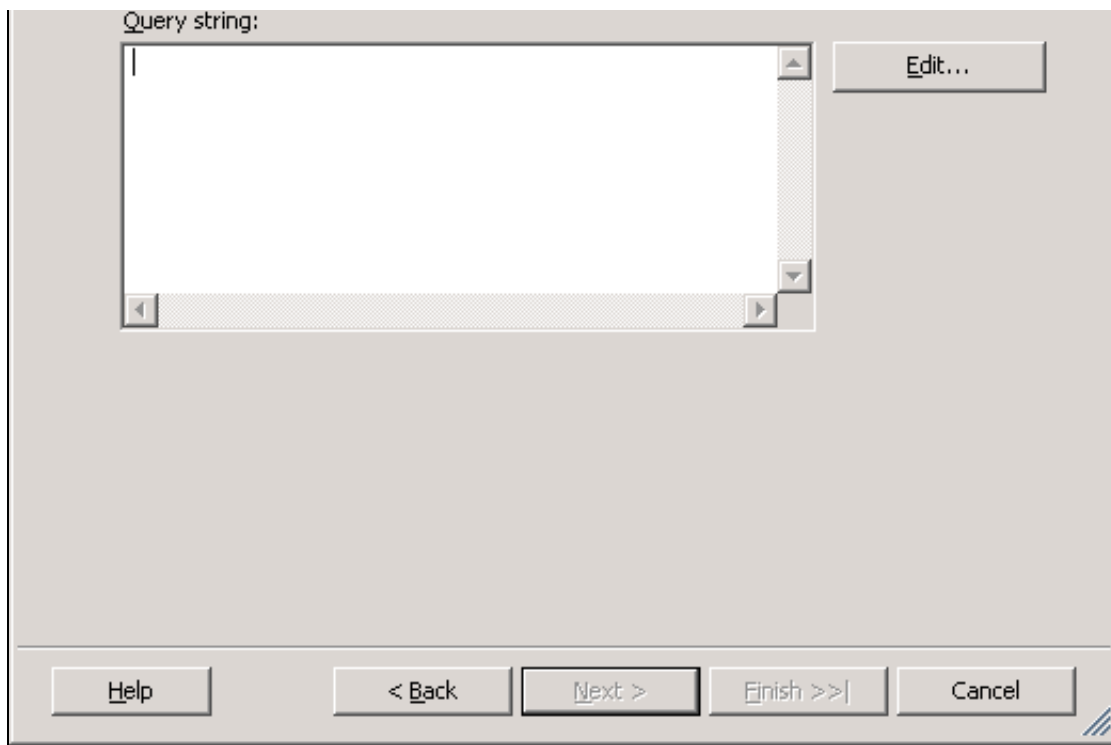
1. Select the server name, or you can type it in.
2. For database Authentication there are two options:
 - "Use Windows NT Integrated security," which means your current logon credentials are passed along behind the scenes to the database, or
 - "Use a specific user name and password," which means your report will have a username and password will be "hard coded" into your report. For this example you should be able to use the first option (Use Windows NT Integrated security) since it is just the Northwind database we will be using.
3. Select or type in a database (e.g. Northwind.)

Click 'Test Connection' to confirm everything has been set up correctly. If so click OK, which returns you to the Report Wizard. Check the "Make this a shared data source" box to reuse this datasource for subsequent reports. With the datasource established, we can now create a dataset for the report to use. Click 'Next.'

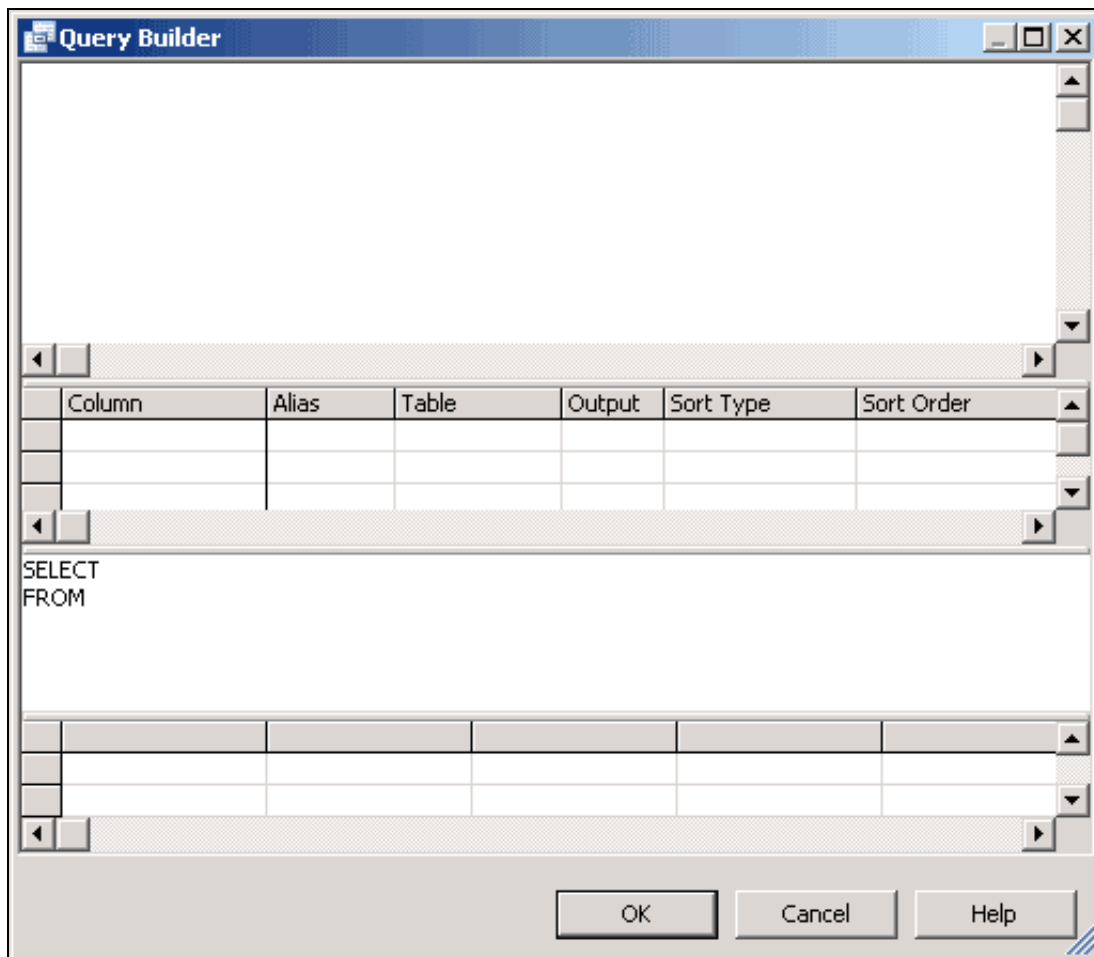


Here you can type or copy and paste in a SQL statement. Instead, click the 'Edit...' button to use the query building wizard.



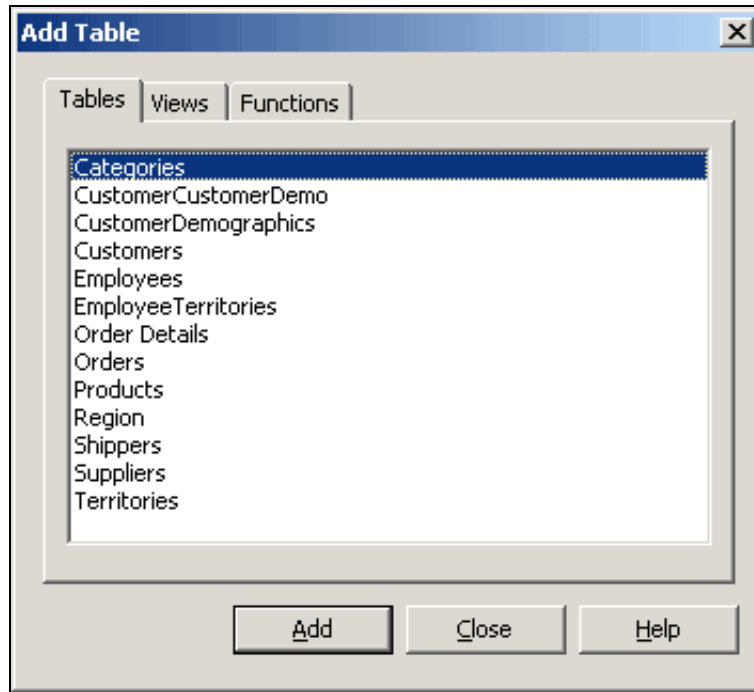


The top part is the graphical list of tables. Below that you specify the columns (what is displayed, sort order, etc.) Beneath that is the text of the SQL statement being generated, and finally at the bottom is the resultset that your query generates. To add a table, right-click in the whitespace at the top and select "Add Table..."

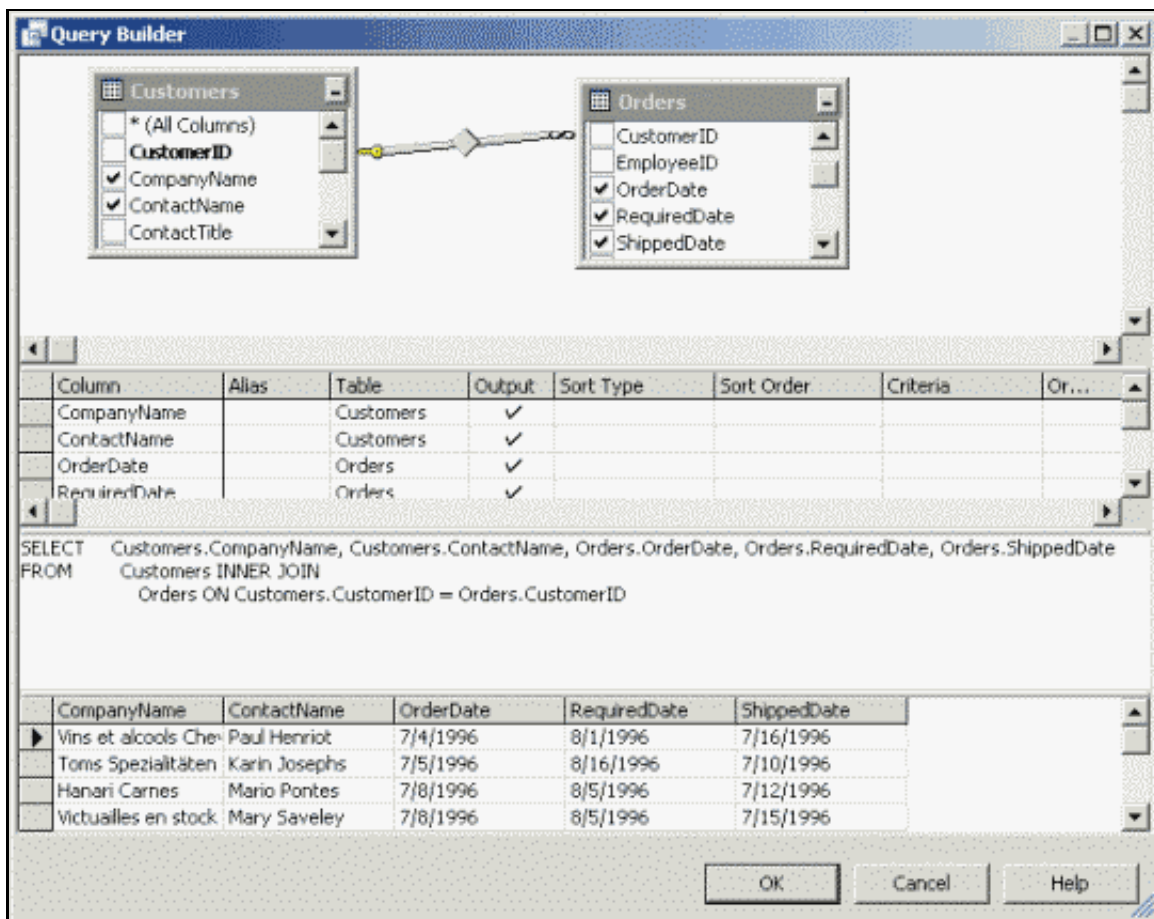


Select the tables you want to use and click Add. For this example add the Customer and Orders tables, then

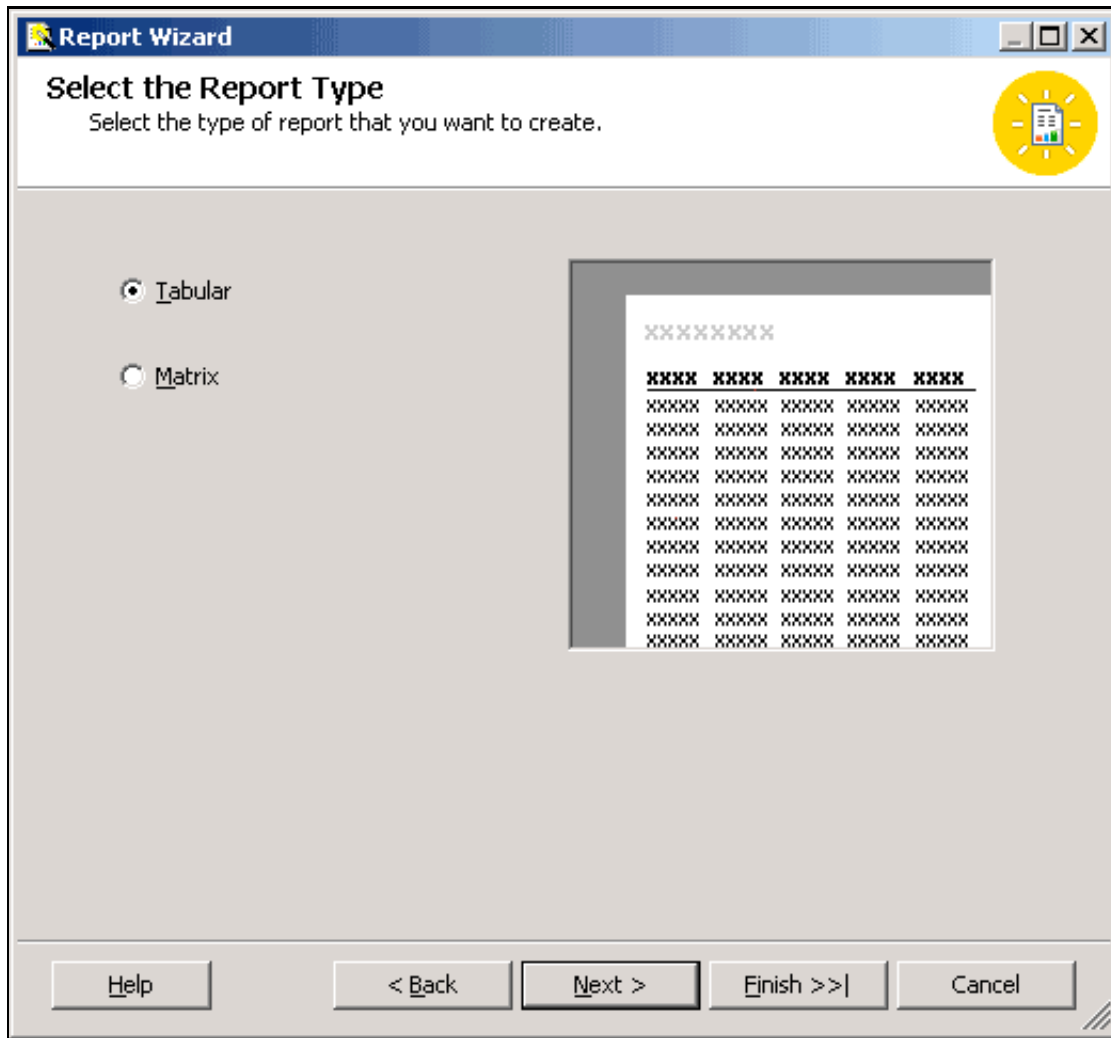
click Close.



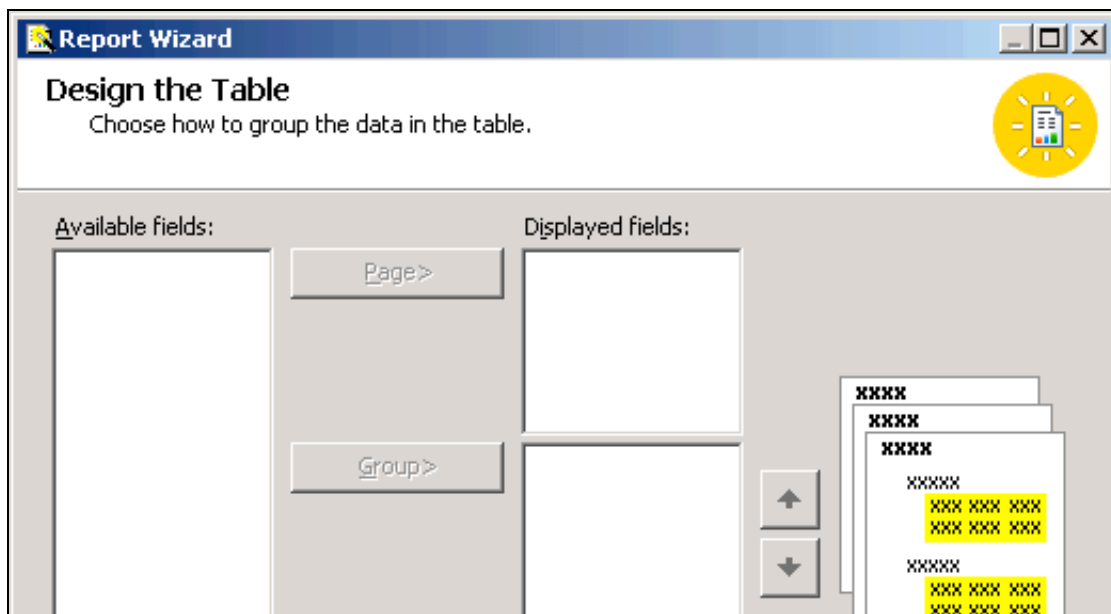
Notice that table relationships that are set up in SQL Server are automatically carried over into the Query Builder. Check some columns to display, then right-click in the table whitespace and select "Run." The results are displayed in the grid in the bottom part of the window. When you are satisfied with your dataset, click OK. The SQL statement you have just built should now be displayed in the Query string window. Click Next.

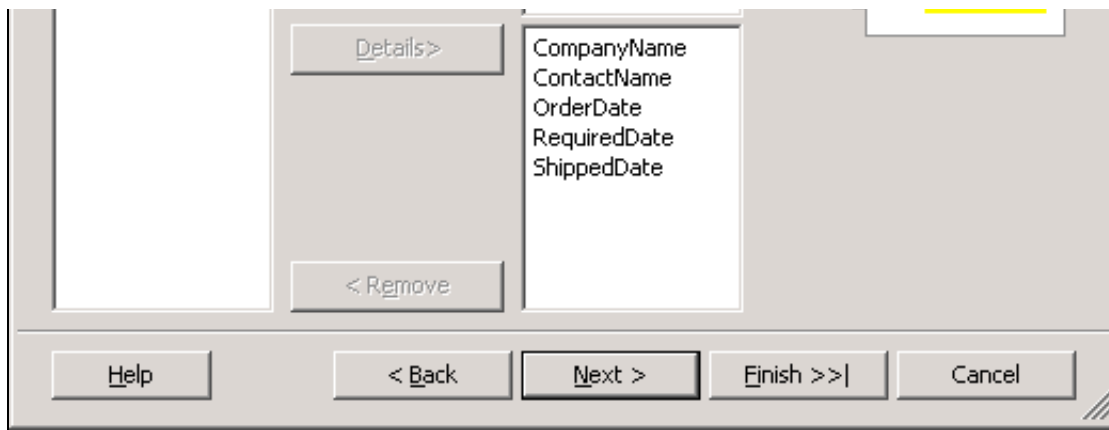


Select the Report Type. The default Tabular format is a basic row-by-column presentation, which is fine for now. Click Next.

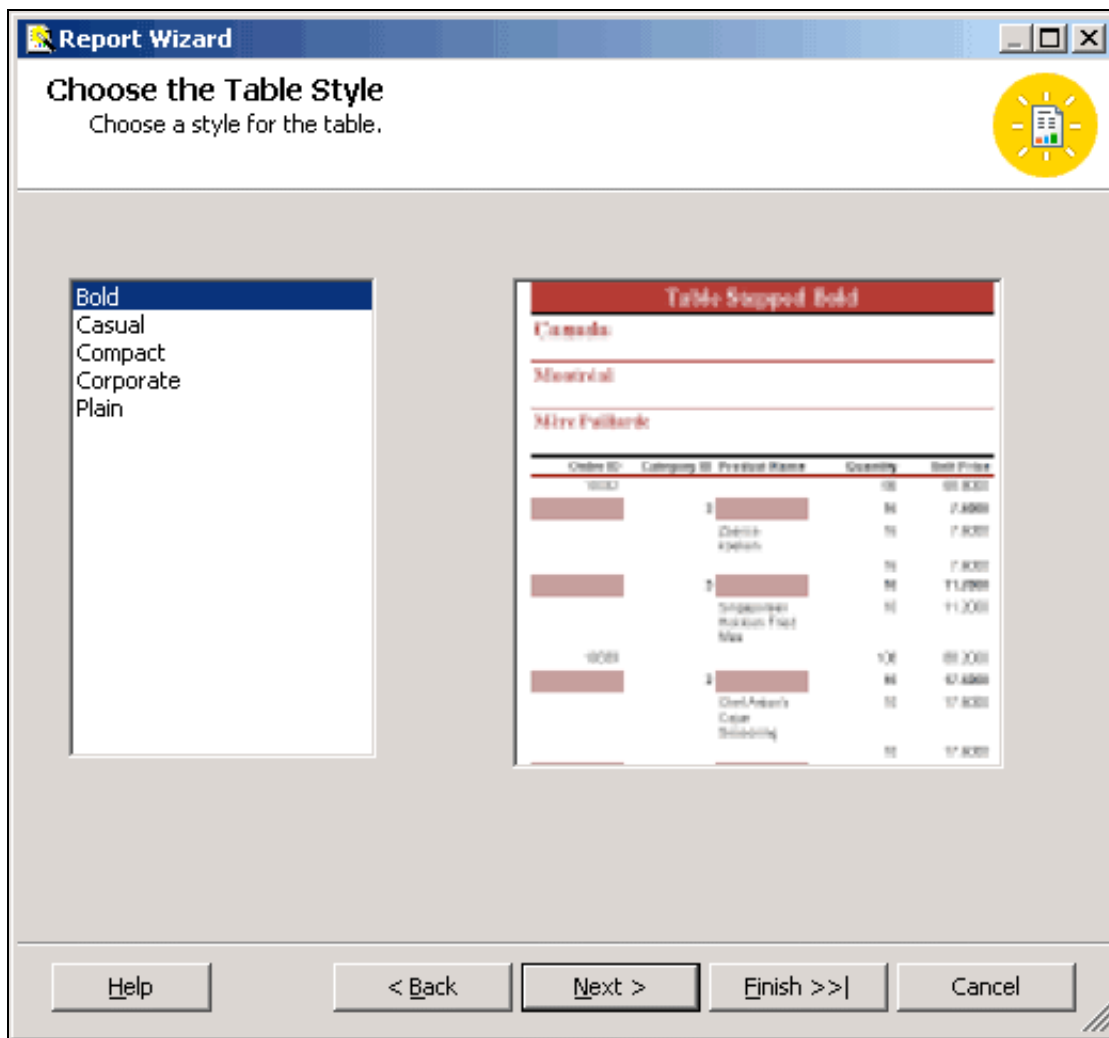


Design the Table. The Report Wizard has some great options for rolling up, grouping, and summarizing your data. But we're going to keep it simple for this tutorial, so select all the fields listed in Available Fields then click 'Details' to move them to the Details box. Click Next.



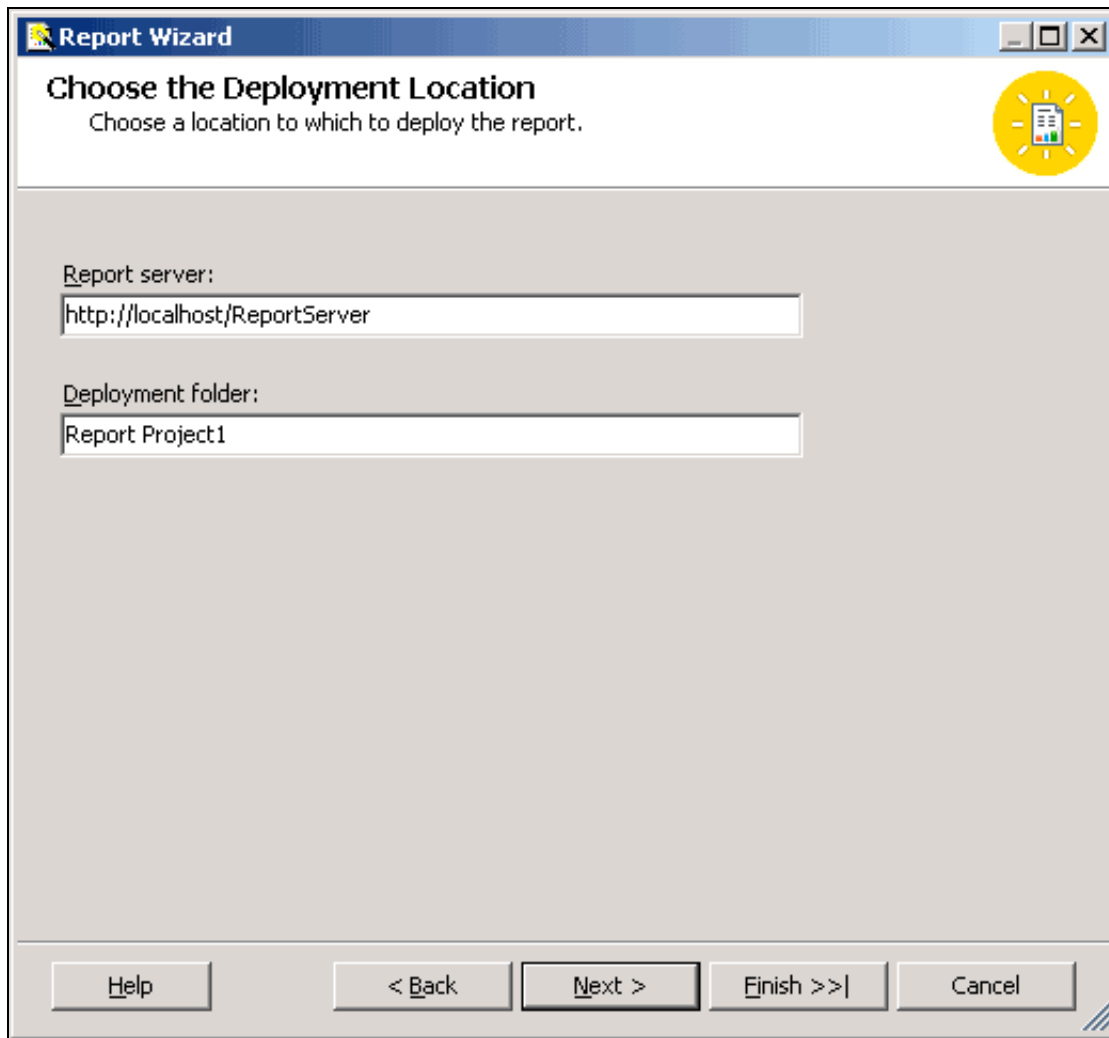


Choose the Table Style. You don't have many templates to choose from now. I have read that Microsoft is working on offering more, as well as a way to create custom templates. For the current version of Visual Studio, however, you only have this paltry list to select from. There is a work-around, however: create a blank report and customize it, then copy and paste this blank-but-formatted report as a new report. The downside of this is that this bypasses the Report Wizard, so you have to build each report from scratch. For this tutorial, keep the Bold table style selected and click Next.



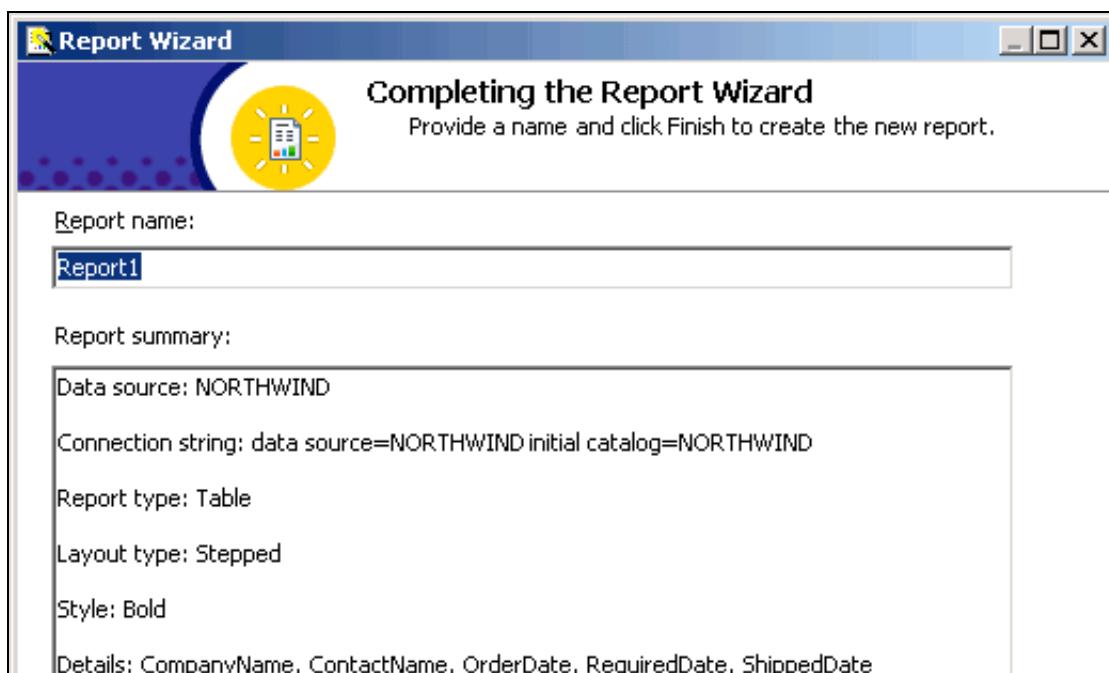
This is where you specify the Report Server to which your report will be deployed. The default is localhost, which is correct if you have SQL Server with Reporting Services running on your local PC. If you have a particular Reports Server on your network in mind, however, type in that server name instead (e.g. <http://myReportServer/ReportServer>). You can also change the name of your deployment folder to

better organize your reports. Once you've set your server name, click Next.

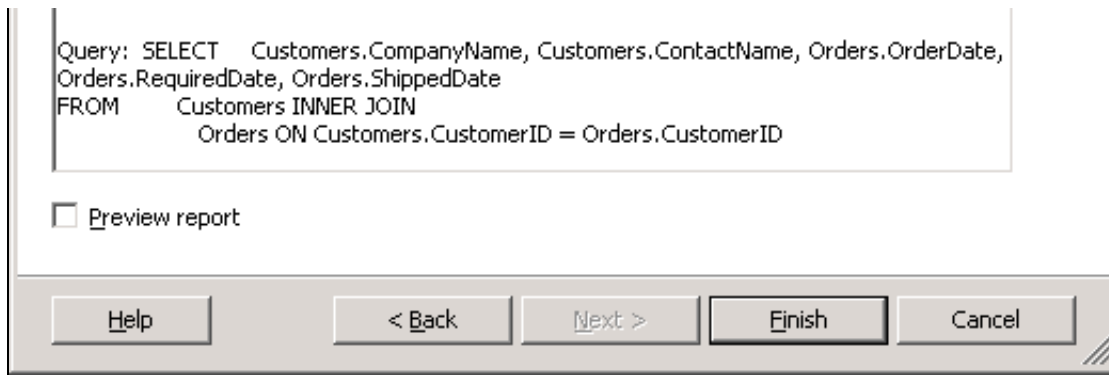


The screenshot shows the 'Report Wizard' window with the title 'Choose the Deployment Location'. The subtitle reads 'Choose a location to which to deploy the report.' Below this, there are two text input fields: 'Report server:' containing 'http://localhost/ReportServer' and 'Deployment folder:' containing 'Report Project1'. At the bottom, there are five buttons: 'Help', '< Back', 'Next >', 'Finish >>', and 'Cancel'.

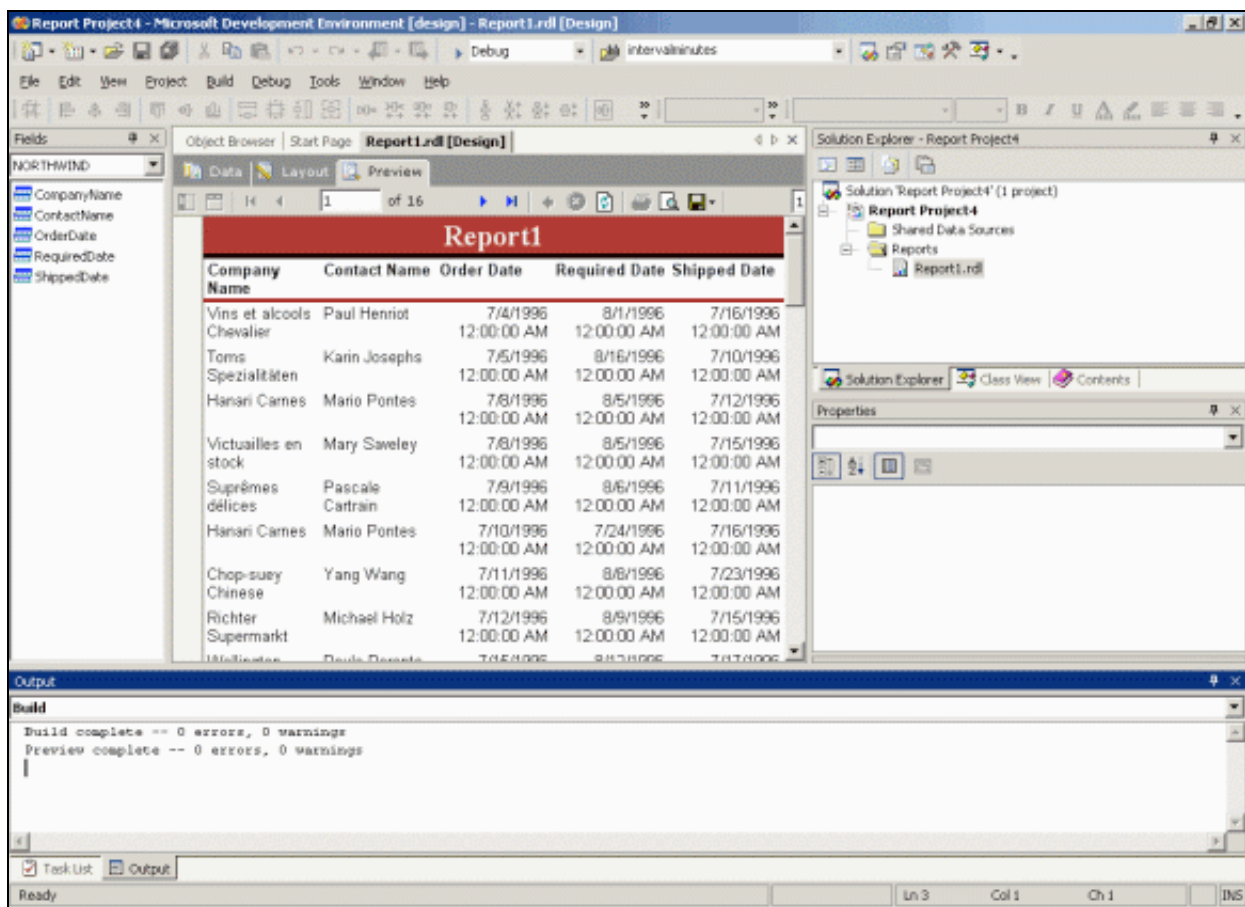
This is the final screen of the Report Wizard, a summary of what you have created. Enter what you would like to name your report at the top. You can also check Preview report to be automatically taken to a preview of your report once you click Finish.



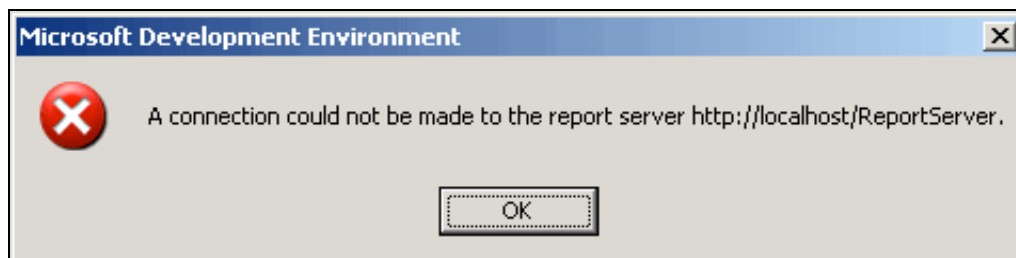
The screenshot shows the 'Report Wizard' window with the title 'Completing the Report Wizard'. The subtitle reads 'Provide a name and click Finish to create the new report.' Below this, there is a text input field for 'Report name:' containing 'Report1'. Underneath is a 'Report summary:' section with a text area containing the following details: 'Data source: NORTHWIND', 'Connection string: data source=NORTHWIND initial catalog=NORTHWIND', 'Report type: Table', 'Layout type: Stepped', 'Style: Bold', and 'Details: CompanyName, ContactName, OrderDate, RequiredDate, ShippedDate'.



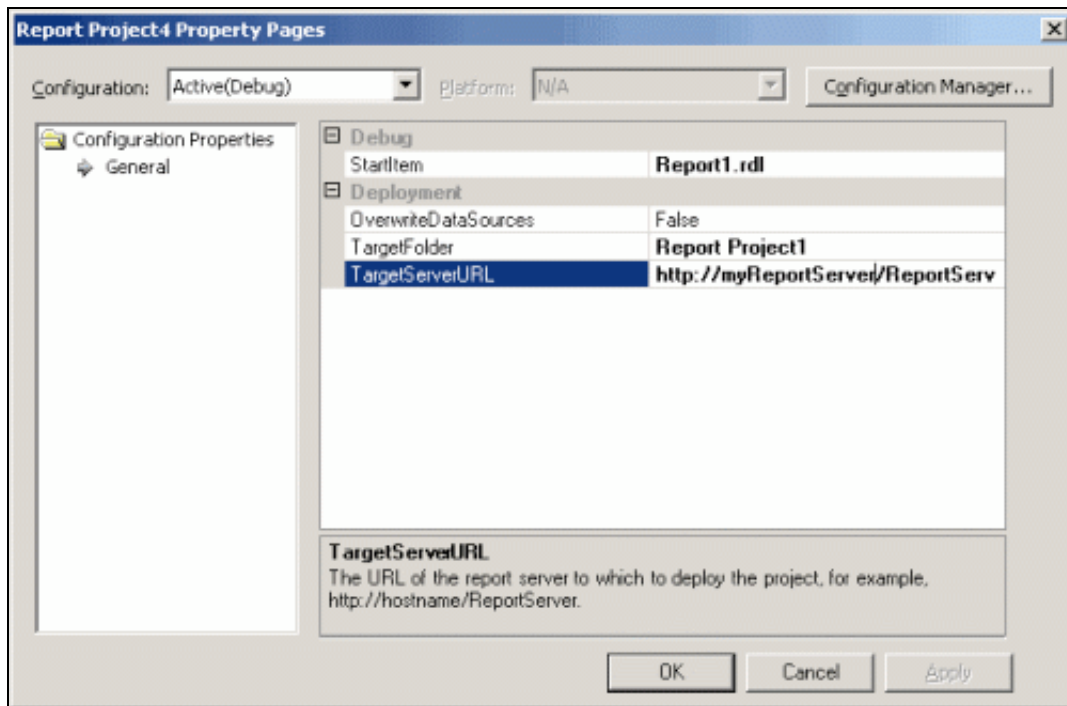
You should now be at the report designer. If you had checked Preview report, you should see just that, with three tabs at the top of the center pane: Data, Layout, and Preview. The Layout tab is where you can fix up the appearance of your report (colors, fonts, grid lines, etc.) No one can view the report until you deploy it to a Report Server, however. So to deploy it, right-click on your report in the upper-right pane and select Deploy.



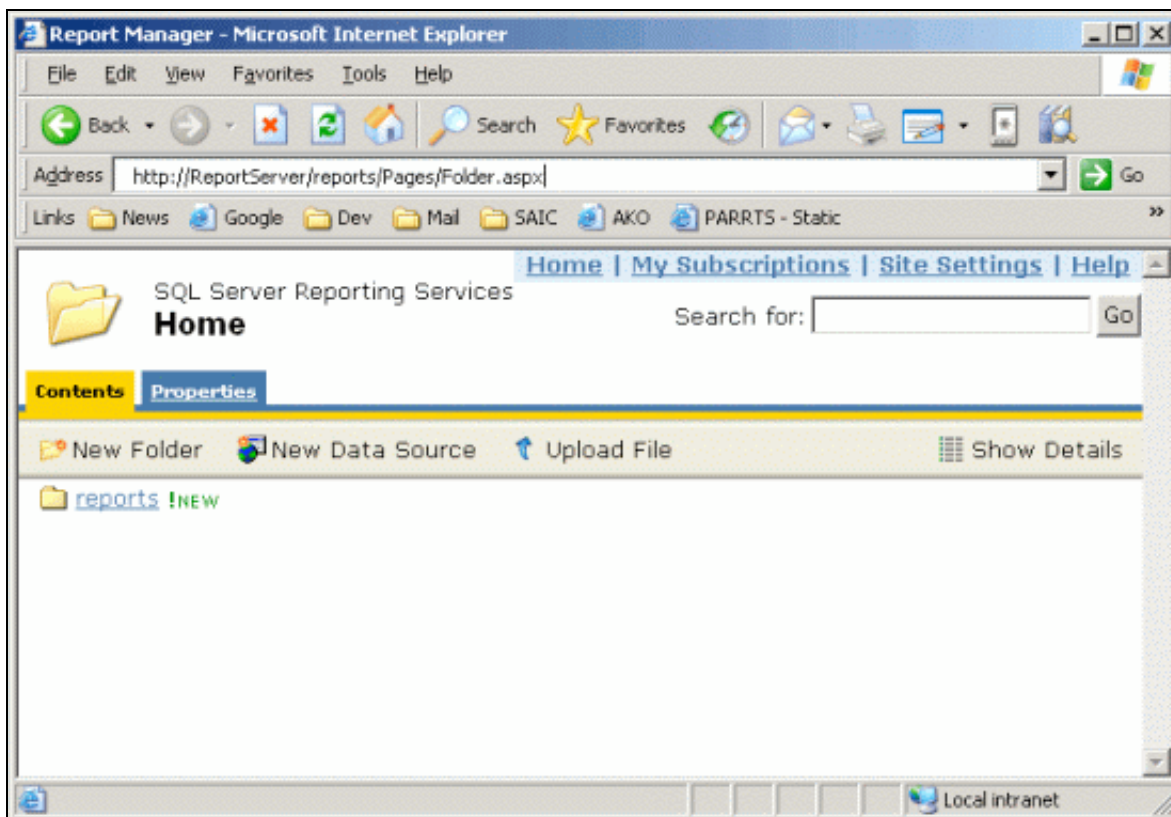
You will get an error if your server is not identified properly, however.



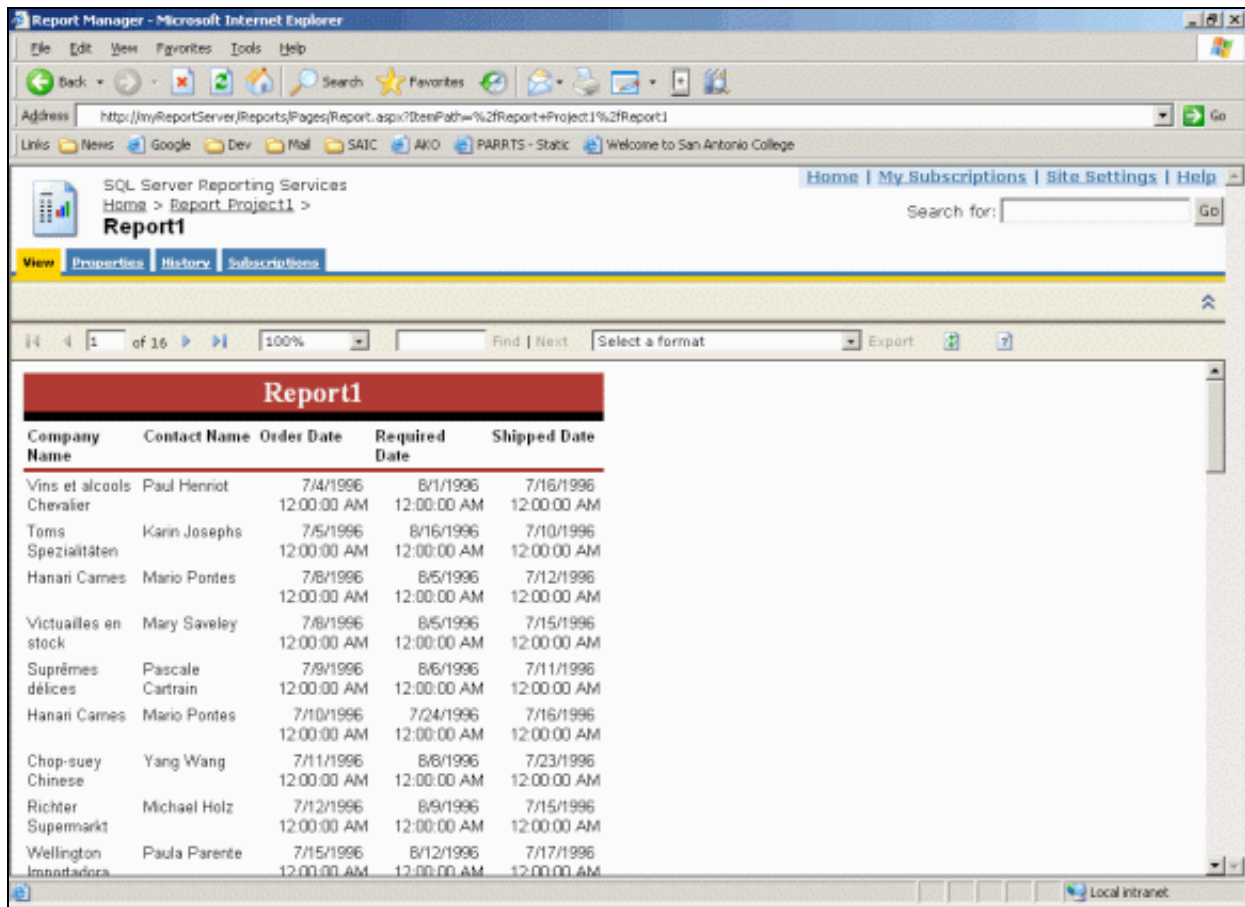
To correct this, select Project | Properties and correct the TargetServerURL field. You can test the server URL by using Internet Explorer. In Internet Explorer, type the address of the report server. For this tutorial, it should be as simple as something like `http://myReportServer/reports/`, where myReportServer is the actual name of the server. If the address works in Internet Explorer you should get the main reports page, which lists the project you just created with a "New!" icon next to it. Click on this project folder, then click on the linked report name to view the report.



By being deployed, the report is now available to everyone on your network. There are security settings available to restrict access to your report (which is beyond the scope of this article.)



To let others know about your new report, give them the URL of the report (copy and paste from the Internet Explorer URL field.) You can also put this URL into a hyperlink to make it linkable by a web page. The report URL will be something like `http://myReportServer/Reports/Pages/Report.aspx?ItemPath=%2fReport+Project1%2fReport1`



Another cool thing about Reporting Services is that it offers several export options on the web. Export options available to the viewer can be restricted by editing the `rsReportServer.config` file. This is just one of a multitude of customization options than can be done by tweaking the configuration text files. The reports themselves (`rdl` files) are actually in XML, so you are not restricted to editing using the Visual Studio GUI. Reporting Services is both powerful and flexible.

Conclusion

Hopefully this article has served as a good introduction to Microsoft Reporting Services. For someone like myself who has spent years hand-coding HTML reports, and suffered with expensive third-party reporting products, this is a welcome solution.

Happy Programming!

• By [Don Franke](#)

About the Author

Don Franke is a contractor and writer currently residing in San Antonio. He can be reached at donfranke@satx.rr.com.

JupiterWeb networks:



Search JupiterWeb:

Jupitermedia Corporation has four divisions:
[JupiterWeb](#), [JupiterResearch](#), [JupiterEvents](#) and [JupiterImages](#)

Copyright 2005 Jupitermedia Corporation All Rights Reserved.
[Legal Notices](#), [Licensing](#), [Reprints](#), & [Permissions](#), [Privacy Policy](#).

[Jupitermedia Corporate Info](#) | [Newsletters](#) | [Tech Jobs](#) | [E-mail Offers](#)