

# **An Introduction to SQL Server Reporting Services**

If you have been doing web application development or design for any amount of time, you have almost assuredly been asked to create reports. Maybe you have created dashboard reports that show very high-level snapshots of productivity or other key measures. Perhaps you have created drill-down employee utilization reports so that managers can see how employees are spending the majority of their time. Or maybe Human Resources has needed to create summary reports for annual performance reviews. Whatever the reason, as a developer, creating data reports has probably been required of you.

To this end, you have many options. You can obviously create a custom reporting solution using something like C# and its related controls to provide the intended information. And depending on the size and requirements of the data, this might be a very attractive solution.

However, what if the report features a large amount of data spanned over an extensive amount of time? And what if management wants drill-down capabilities or, even worse, graphical representations such as charts and graphs? While it is certainly possible to accomplish this in C#, it also adds a level of complexity that many developers either do not want to deal with or, quite frankly, do not have the time to deal with.

To address this need, several server-side reporting technologies have been introduced over the years. If you have been in the developer community long enough, you have probably heard of some of the big names such as Oracle Reports or Crystal Reports. In the last few years, however, Microsoft has introduced its own server-based report generation software: SQL Server Reporting Services (SSRS). While there are plenty of similarities between SSRS and its competitors, SSRS has at least one glaring advantage over any other game in town; it is free. Well, sort of. It is tightly

# Chapter 1: An Introduction to SQL Server Reporting Services

---

integrated within the SQL Server infrastructure, and if you have SQL Server installed, you can also have SSRS for free. (The *sort of* part of *free* is that you have to have pay for SQL Server, obviously. But once you do, SSRS is free.)

If you grasp the importance of reporting in a business environment, you can probably see how valuable this technology can be within a SharePoint installation. After all, SharePoint is first and foremost a web portal and, as such, is full of reportable data. Whether the data consists of tasks assigned to employees in a given time span or the number of active discussion boards employees participated in this month, there is plenty of data for any SharePoint installation. But beyond that, it is also fair to state that many organizations that deploy SharePoint have other business data they would like to report within SharePoint. Maybe they have an inventory and sales system used for tracking the ongoing business of their operations. Wouldn't it be great to be able to report this data within SharePoint without having to import the data directly into your SharePoint lists or libraries?

However, integrating SSRS and SharePoint provides some unique challenges and architecture. This book will walk you through these considerations and give you a good idea of how to best use these two technologies to make awe-inspiring reports that will accurately provide the information that decision makers want and will even allow decision makers to easily manipulate the parameters of the reports to meet their needs.

This chapter will start you on your journey by giving you an understanding of exactly what SSRS is and providing an overview of some of the obstacles you will face as you develop reports in your SharePoint environment.

And with that, it's time to get started.

## What Is SQL Server Reporting Services (SSRS)?

SQL Server Reporting Services (SSRS) is a server-based reporting platform that allows you to deploy professional-looking and powerful reports fairly easily (once everything is set up). You get the drag-and-drop functionality that you may be familiar with if you have used any version of Visual Studio over the years. You can literally drop a chart onto your report, drag some data fields to the chart, and then deploy the report. You can also do more powerful things such as create pie charts where each pie segment, when clicked, takes the user to a related matrix of data where they can drill down even further to find more detail.

One of the nicer details about SSRS is that it will allow you, as a developer, to access a variety of data sources to feed the report. Of course, since it is a SQL Server product, you can easily tie into your SQL Server database. But beyond that, you can connect to a whole slew of other data objects, as seen in the following table (obtained from <http://msdn.microsoft.com/en-us/library/ms159219.aspx>):

Source of Report Data	Reporting Services Data Source Type	Version of Data Source
SQL Server relational database	Microsoft SQL Server	SQL Server 2008 SQL Server 2005 SQL Server 2000 SQL Server 7.0
SQL Server relational database	OLEDB	SQL Server 2008 SQL Server 2005 SQL Server 2000 SQL Server 7.0
SQL Server relational database	ODBC	SQL Server 2008 SQL Server 2005 SQL Server 2000 SQL Server 7.0
Analysis Services multidimensional database	Microsoft SQL Server Analysis Services	SQL Server 2008 Analysis Services SQL Server 2005 Analysis Services SQL Server 2000 Analysis Services
Analysis Services multidimensional database	OLEDB	SQL Server 2008 Analysis Services SQL Server 2005 Analysis Services SQL Server 2000 Analysis Services
XML	XML	XML Web Services or documents
Report Server Model	Report Model	Models can be created from: SQL Server 2008 SQL Server 2005 SQL Server 2000 SQL Server 2005 Analysis Services Oracle 9.2.0.3 or later Teradata v12 and v6.2
SAP multidimensional database	Sap BI NetWeaver	SAP BI NetWeaver 3.5
Hyperion Essbase	Hyperion Essbase	Hyperion Essbase 9.3.0
Oracle relational database	Oracle	Oracle 10g, 9, 8.1.7

(continued)

## Chapter 1: An Introduction to SQL Server Reporting Services

---

Source of Report Data	Reporting Services Data Source Type	Version of Data Source
Oracle relational database	Part of OLEDB	Depends on version of MDAC
Teradata relational database	Teradata	Teradata v12 Teradata v6.20
DB2 relational database	Customized registered data extension name	
Generic OLEDB data source	OLEDB	Any data source that supports OLEDB
Generic ODBC data source	ODBC	Any data source that supports ODBC

When you see things like “Any data source that supports ODBC,” the field is pretty wide open. Any data source that supports ODBC? That is pretty encompassing.

One of the more interesting items on this list is the XML data source. This data source allows you to connect to pretty much any web service out there (given that you have the correct connection information and credentials). So anything not in this list? Well, you can create a web service on your server, grab the data you need, and send it back in a DataTable or DataSource object, and reporting services can read it. That is pretty powerful.

### ***Is SSRS just Microsoft’s Version of Crystal Reports?***

When looking at SSRS, it is not difficult to begin drawing comparisons to other similar products, especially something like Crystal Reports, if you have ever worked with those technologies in the past. They are very similar, at least as far as they both are tools that allow you to create reports on your business data.

So which is better? Has more features? Is easier to use? That probably depends on who you ask. If you do a Live Search for something like “SSRS versus Crystal Reports,” you will find page and page of hits and, depending on what camp the author comes from, the winner will be different on every page. Many people will say Crystal Reports is more powerful; others will argue that SSRS has better handled the primary functions of the reporting interface. Most people will probably say it also depends on how you plan to deploy your reports (web reports or standalone). However, at least in the world of SharePoint, SSRS wins.

For one thing, if you are using SharePoint, you already have access to SSRS. SSRS came as a free add-in for SQL Server 2000 and has been included with both SQL Server 2005 and SQL Server 2008. So, since you are most likely using a SQL Server technology for your data backend for SharePoint, it is a fair bet that you already have SSRS, even if you don’t have it set up yet.

# Chapter 1: An Introduction to SQL Server Reporting Services

Additionally, SSRS has two specific modes that can be installed: Native and SharePoint Integrated Mode. That's right; one of the two SSRS modes that can be installed is specifically targeted toward SharePoint. Installing SSRS in this manner, as you will see in Chapter 2, will give you access to special Web Parts specifically meant to handle SSRS reports. It will also allow you to deploy your reports directly to libraries in your SharePoint site collection (as opposed to deploying them directly to the Reporting Services server).

Finally, another huge advantage is that, if you are a SharePoint developer (or any .NET developer, for that matter), you are already intimately familiar with the SSRS designer, which is called the Business Intelligence Development Studio (BIDS). This is because BIDS is simply an add-on to Visual Studio 2008, and all development work, as such, is done within the Visual Studio application. So when you begin development of your SSRS 2008 reports, you will be designing them within Visual Studio 2008, which you probably are already familiar with. To verify this, you need only to glance at Figure 1-1.

*You will get an overview of BIDS, including where to get it and how to install it, in Chapter 4.*

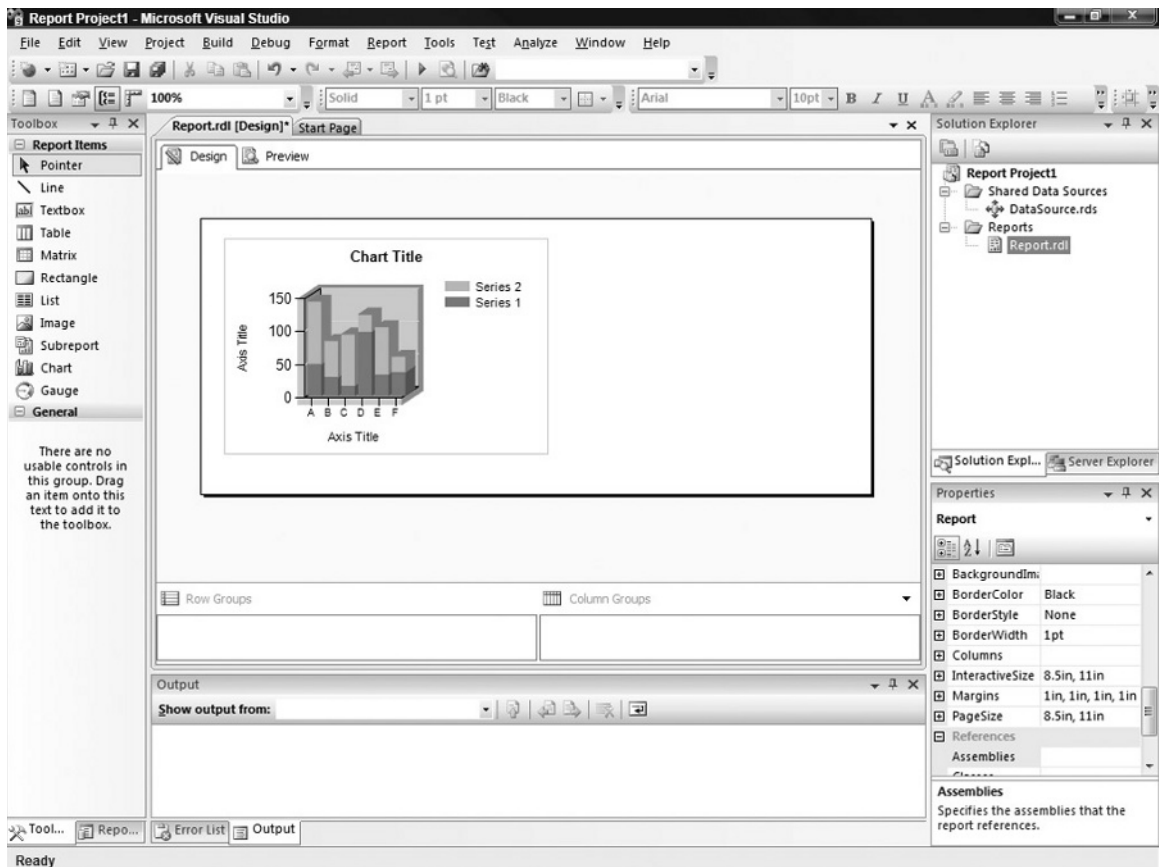


Figure 1-1

# Chapter 1: An Introduction to SQL Server Reporting Services

It is worth mentioning at this point that there is another option for developing SSRS 2008 reports: the Microsoft SQL Server 2008 Reporting Services Report Builder 2.0. You can download this product from the following location: <http://www.microsoft.com/downloads/details.aspx?FamilyID=9f783224-9871-4eea-b1d5-f3140a253db6&displaylang=en>. Alternatively, you could just use the following shortened reference to the same URL: <http://tinyurl.com/ReportBuilder2-0>.

This application, also free, is a standalone report designer that has all of the same features of BIDS in an arguably better Microsoft Word-like format, as can be seen in Figure 1-2.

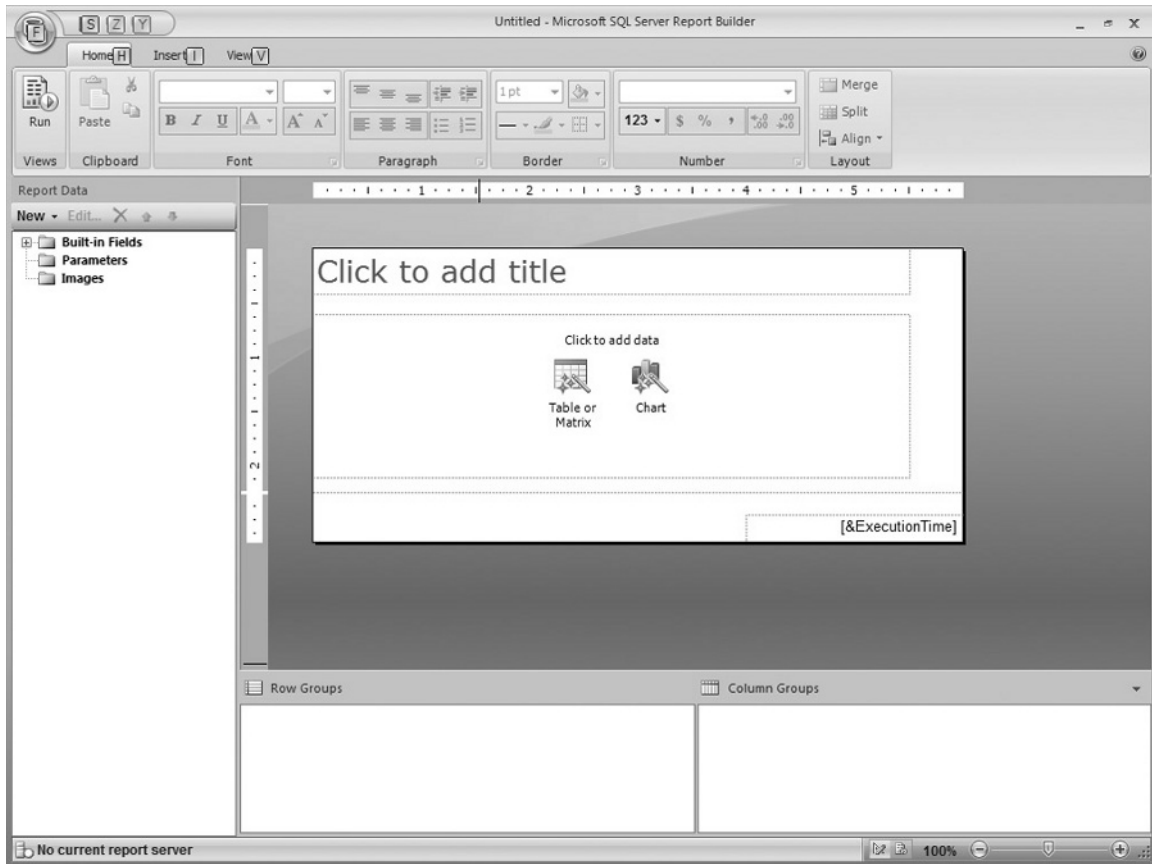


Figure 1-2

While this might be a viable option for your own report development and certainly warrants your own evaluation, it will not be used for this book. Since BIDS is just using the Visual Studio 2008 shell, it will be familiar to a lot of SharePoint developers. And even if you are not a SharePoint developer, there is still a chance you have used Visual Studio in your experience in development. So the interface won't be entirely new. Although Report Builder might work for you, it would be easier to explain BIDS within the

Visual Studio context than it would be to explain an entirely new concept. And if you are already using Visual Studio, you will probably be more comfortable staying there. Thus, it isn't really feasible to show how to create reports generated in both platforms throughout the entire book; using only one platform is more practical, and BIDS wins out. Much of the planning and development you do in BIDS, however, will transfer (knowledge-wise) in your future attempts to use Report Builder if you should so choose.

### **What Are Some of the New Features of SSRS 2008?**

As with any new software release, SSRS 2008 has some very cool new features that appeal not only to the developers who create the reports but to the end users who will use them in their decision-making strategies. These include things like enhanced rendering extensions and graphs for the end user, as well as a simplified development experience with better integration to your current server infrastructure.

#### **A Better User Experience**

One of the most discussed new features of SSRS 2008 is the incorporation of Dundas charts bundled as part of the development environment. These include both radial and linear graphs, as well as some pretty neat "slider" charts. You will see a lot more of these charts in later chapters but, to give a small tease, look at a simple dashboard report that can now be easily created using the new Dundas Radial chart as seen in Figure 1-3.



Figure 1-3

While these new charts and graphs are impressive on their own, it is also worth noting that Microsoft has added 3D rendering to all of its previously included charts. For example, in SSRS 2005, you could produce an exploded pie chart similar to the one shown in Figure 1-4.

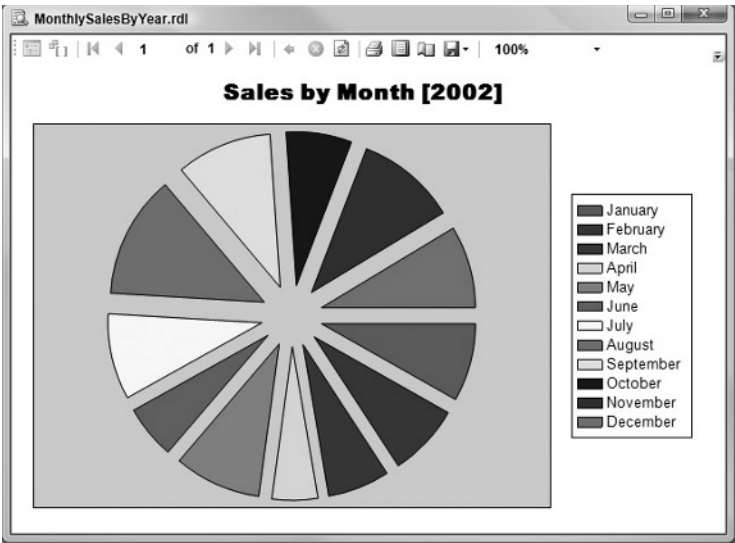


Figure 1-4

However, with the exact same data and with no additional formatting (i.e., using the default settings), you can now also produce a 3D version of the exploded pie chart, shown in Figure 1-5.

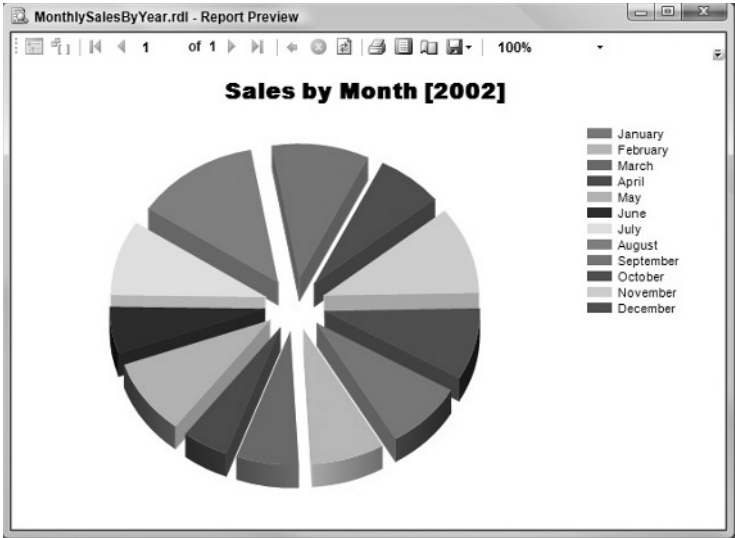


Figure 1-5



While the difference may be hard to see in the black-and-white constraints of this manuscript, it will become very apparent in the charts you create. Along with the Dundas gauges, your reports will have a level of elegance and professionalism not available out of the box in earlier versions.

In addition to all of the enhanced graphs and charts discussed in this section, it is also worth noting that reports can be exported in several different formats. As has been possible since at least the SSRS 2005 release, you can export your report directly from the web version to formats such as PDF, Excel, and CSV. With SSRS 2008, however, you can also export directly to Microsoft Word. This provides a new level of flexibility for generating and distributing your reports to those who need them.

As a final note with regard to a better user experience, users now have better integration with their SharePoint environment for reports deployed in SharePoint integrated mode (you can read more about SharePoint integrated mode in Chapter 2). For example, users can now create data-driven subscriptions for their reports deployed to a SharePoint library.

While there are several other features that directly apply to the end user, the ones highlighted in this section are surely some of the biggest and will get users excited about upgrading to the new SSRS 2008.

## ***A Simplified Development Experience***

While there are several new developer enhancements in SSRS 2008, the biggest one is probably the fact that SSRS no longer requires Internet Information Services (IIS) to run. And yet, even with this being true, none of the functionality of previous versions has been lost. This is done through the use of native support for HTTPS and ASP.NET; URL management for site and virtual directory names; a new authentication layer; and health monitoring through new memory-management features. This is a pretty impressive change in the architecture of SSRS and will get a lot of developers buzzing.

Perhaps the most exciting part of diving into the developer experience in SSRS 2008 for the first time is that once you get started you will feel at home, especially if you are used to creating web applications using .NET technologies. If you are used to creating your web application pages in Visual Studio, guess what? You will be creating your reports in Visual Studio. If you are used to modeling your data backend in SQL Server, you can still do that, too. Maybe you are used to accessing web services to mash up data to display on your pages. You have the power to do that in SSRS as well.

While this is still going to be a little intangible this early in the book, if you have done web application development using tools like Visual Studio and SQL Management Studio, you will find yourself thinking “hey, I’ve done something very similar to this in the past,” or “this is much easier than I thought it would be,” several times as you read the following chapters.

However, at the same time, you will be able to tap into a power that you generally don’t get in your other applications. You will have a fairly easy way to create eye-catching graphical interfaces that pretty much all management loves to see. You will be able to take your data and, rather than just report it, you can showcase it. You will find how easy it is to create dazzling reports with simple drag-and-drop controls and intuitive designers. You will take what is familiar and go with it to a new and exciting place.

While the end users will get a lot out of the reports and the managers will certainly enjoy the end product, the development experience will be one that is familiar and, hopefully, a lot of fun to work with.

# SSRS Deployment

While this will be covered in much more detail later in this book, deployment warrants at least a mention in this introductory chapter, as it is a crucial part of the reporting process. After all, if your users can't get to the reports, what value do they have?

When SSRS is set up in SharePoint integrated mode (see Chapter 2), a couple of things happen. First, you actually deploy your reports and their related data sources directly to SharePoint libraries hosted in your target site collection. This means that, at any given time, you can navigate to your site, find your reports library, and see a listing of the currently deployed reports. You can even run the reports directly from the library simply by clicking them.

Another thing that happens is that you get new Web Parts installed on your server that will be available to all site collections that utilize SSRS. These Web Parts will allow you to set up a report to be hosted directly on any page within your site collection. This way, the reports will look as though they are a seamless part of your existing page, as shown in Figure 1-6.

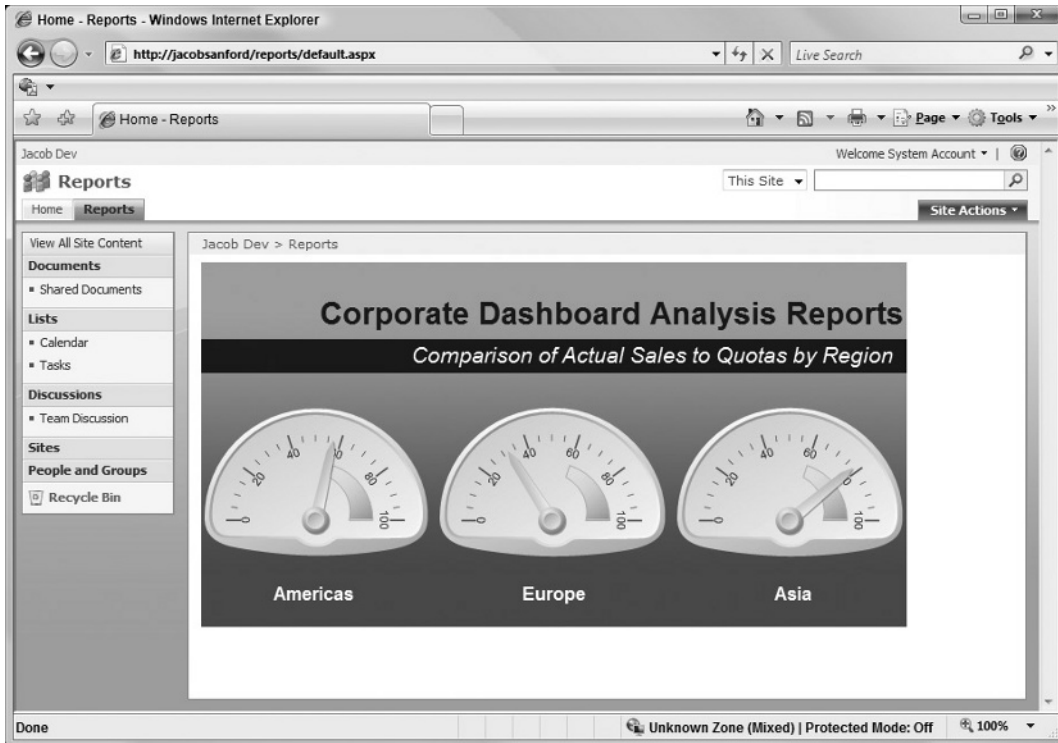


Figure 1-6

### Summary

This chapter has provided a basic overview of SSRS and the new features available in SSRS 2008. You have seen the basic development tools available to create new reports and some of the enhancements to the user and development experience that make SSRS 2008 much better than any of its predecessors. You have also been given a brief glimpse of how your reports integrate with your SharePoint installation and how your reports will look and feel once deployed to your site collection.

This chapter was meant to provide an SSRS 2008 and SharePoint integration. It was meant to foreshadow a bit for future chapters and to offer at least a cursory understanding of SSRS so that subsequent chapters will be more accessible. And with that it is time to get to the fun stuff: making reports.

