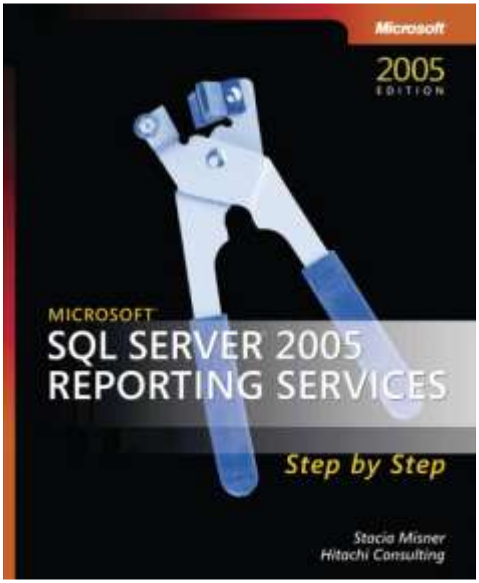


Microsoft® SQL Server™ 2005 Reporting Services Step by Step

Stacia Misner; Hitachi Consulting

To learn more about this book, visit Microsoft Learning at
http://www.microsoft.com/mspress/books/9153.aspx

9780735622500
Publication Date: April 2006

Table of Contents

Part I Getting Started with Reporting Services

Part II Authoring Reports

Part III Managing the Report Server

Part IV Delivering Reports

Chapter 1

Understanding Reporting

After completing this chapter, you will be able to:

- Understand the purpose of enterprise and ad hoc reporting.
- Recognize the characteristics of a reporting platform.
- List the constituents of reporting user communities.
- Describe the stages of the reporting life cycle.
- List the features and components of Microsoft SQL Server 2005 Reporting Services.

In this chapter, you see what enterprise reporting and ad hoc reporting are all about, as well as how they differ from other types of reporting. You also review how different groups within your organization need to use or support reporting and how they participate in the reporting life cycle. With this foundation, you'll better understand how the various components of Reporting Services fully support the reporting needs of your organization.

Reporting Scenarios

Since you're reading this book, it's likely that you work for a company that needs to be able to share information. Whether your company is small or large, can you imagine what would happen if employees couldn't access the information they need to do their jobs? The decisions that each individual employee makes during the course of his daily tasks have a profound impact on successful business operations and rely on easy and regular access to information.

One way that a company commonly shares information is through enterprise reports. For the purposes of this book, an *enterprise report* is considered to be the presentation of information that is formally distributed to some or all individuals across an enterprise, or even to individuals outside the enterprise. This information can be presented in a variety of formats, for example, as a Microsoft Excel spreadsheet or a text document. It can also be delivered as a printed report or sent to a list of recipients as an e-mail attachment. Information can also be made available in a central location, such as on a Web page on a corporate intranet or embedded in a portal where users can access reports when needed.

Sometimes a company needs a less formal way of creating and sharing information, especially if business dynamics are changing rapidly and there's a need to provide quick access to current information to a small group. In this situation, it's less important to adhere to presentation standards, yet the quality of information must be consistent with traditional enterprise reports. These *ad hoc reports* tend to be simpler than enterprise reports in terms of data content and presentation features, and are usually based on standard nontechnical representations, or *models*, of underlying data sources. A business person should be able to create an ad hoc report without knowing how to write a relational query and without having to wait for technical assistance in order to pull together data to answer a new business question.

The diverse types of organizations that use reporting and the differences in their information needs make it difficult to compile a comprehensive list of all possible reporting scenarios. However, you can look at who is using the shared information to develop the following generalizations about reporting scenarios:

- *Internal reporting* is probably the most common enterprise reporting. This category of reporting involves the sharing of information within an organization across all levels of employees and usually involves standard departmental reports. For example, employees in the product warehouse might regularly receive detailed printed order reports every morning. Elsewhere in the company, managers might get a financial statement in an Excel workbook delivered as an e-mail attachment when the books are closed each month.
- *External reporting* can take many forms, but is defined as disseminating information to people outside an organization. This information might be printed and mailed, such as shareholder reports. Increasingly, companies are publishing annual reports as PDFs (Portable Document Format files) for interested parties. External reporting can even include the exchange of information between business systems, such as invoicing information sent to a customer's receivables system electronically.

You can also consider how the information is being accessed to develop the following additional generalizations:

- *Standard reporting* relies on a central storage location that can display a list of contents or a catalog of the available reports so users can find the reports they need. Usually, security is applied to report storage to control the reports that individual users can see. Reports might be organized in a proprietary reporting platform repository or some other type of document management system.
- *Ad hoc reporting* depends on the availability of a model that allows users to select which data elements should be included in a report, along with a designer tool that allows them to arrange the layout of these elements to produce a simple report. These reports might be saved to a central repository to share with others or they might be stored on the user's local hard drive for personal reference.
- *Embedded reporting* is the integration of reports into portals and in-house or third-party applications. For example, many companies are migrating to Web-based line-of-business applications for accounting and payroll functions. Instead of building reporting processes into these applications, these companies can leverage an extensible enterprise reporting platform to allow users to access information by using these applications.

These reporting scenarios have the following two characteristics in common:

- **Central storage** Reports or report models (for ad hoc reporting) are accessed from a central location. Reports might also be delivered directly to users from a central server. Many people need access to the same information, possibly in different formats. Often, access to information needs to be limited to those with a need to know.
- **Standardization** Enterprise reports conform to a standard design with a consistent layout, while ad hoc reports conform to a standard data model to ensure consistent results.

In addition, the proliferation of information that can (or should) be available to the average worker has led to increasingly more sophisticated requirements for an enterprise reporting solution. For example, users need to be able to do the following:

- Navigate easily within a large report.
- Move from one report to another while maintaining context.
- Access previous versions of a report to compare information at different points in time.
- View data consolidated from multiple sources into a single report.

An enterprise reporting solution also needs to satisfy administrative requirements. A reporting platform should have the following characteristics:

- Flexibility to store a single report from which multiple versions may be generated based on changeable parameters or user profiles.
- Ability to support a push-pull paradigm, in which users can seek out the information they need online or subscribe to information that is sent to them on a periodic basis.
- Capability to manage reports using a Web interface so that administrators can perform tasks without being tied to their desks.

Reporting User Communities

Many people within an organization are usually involved in some aspect of reporting. Typically, users are members of one or more of the following communities: information consumers, information explorers, and analysts.

Most users—typically 65–80 percent of the total user population—are information consumers. Information consumers usually view static and predefined reports. If they use printed reports, they might get them the old-fashioned way—someone does a batch print of hard copy reports, then sends it out or delivers it to each recipient's in-box. A more technically oriented environment might make a document repository available, providing the electronic equivalent of a file cabinet that information consumers can access at will. In some cases, information consumers need to receive reports on a recurring basis, such as a weekly update on key performance measurements. Some of these users might want to produce their own simpler reports when an existing report doesn't answer a particular question.

One of the many strengths of Reporting Services is its ability to provide easy access to a wide array of predefined reports, making information consumers a key audience served by Reporting Services. Although many people might prefer to view information online, they can still get printed reports or can get reports delivered via e-mail. In either case, reports can be processed on demand (where information is as current as the data in the source system) or on a scheduled basis (where information represents a specific point in time). For maximum flexibility, an information consumer can choose from a variety of formats that can be delivered to a range of devices. Finally, Reporting Services provides ad hoc reporting capabilities so information consumers can obtain quick answers as needed.

Information explorers typically constitute 15–25 percent of the user population. Like information consumers, they use predefined reports, but they also interact with reports. For example, information explorers commonly use filters to isolate segments of data. Information explorers might also interact with reports by starting with summary information and then moving to more granular levels of detail, whether drilling down to view details in the same report or drilling across to view related information in a separate report.

Interactive reports suitable for information explorers require more work to develop than static reports, but Reporting Services has a wide array of features to support the development of these reports. Parameters can be designed into a report to support filtering data at the source or in the report. An information explorer can change parameter values on demand, or an administrator can predefine specific parameters for different groups of information explorers. Reports can also include dynamic visibility to support drilling down or actions to support drilling across.

The smallest user community, typically representing 5–10 percent of users, includes analysts. This group possesses the skills to develop free-form reports that facilitate complex data analysis. Such reports are often in spreadsheet form; through them, analysts can enhance the data with sophisticated calculations, such as linear regressions and allocations. These reports can eventually be shared with information consumers and information explorers.

Out of the box, Reporting Services supports analytical needs by providing the ability to export a report to Excel. Conversely, an Excel workbook created by an analyst can be uploaded to Reporting Services as a resource to be shared with the rest of the user community. Also, because Reporting Services is an extensible architecture, a custom application or third-party plug-in can give analysts complete flexibility to develop free-form reports within the Reporting Services environment.

The Enterprise Reporting Life Cycle

The *enterprise reporting life cycle* is a three-stage process through which a report progresses from authoring to management to delivery. A reporting platform must not only serve the needs of each reporting community, but must also fully support each stage of the reporting life cycle. It should also provide the architecture, functionality, and utilities to support the activities of authoring, managing, and delivering reports. In other words, everything you need from the beginning to the end of the reporting process should be in one integrated product set. Reporting Services provides just that.

Authoring

The primary activities of the authoring stage include defining the data to be presented in a report, organizing the data into a structured layout, and applying formatting to enhance the report's appearance. For example, when executive management needs to monitor sales performance across product lines, a report author can create either an enterprise report or an ad hoc report to present sales data in a table layout. To facilitate analysis, however, the report author might use conditional formatting to highlight products for which performance exceeds defined performance goals or fails to meet these goals. This advanced formatting technique is an enterprise reporting activity.

To support authoring, Reporting Services provides a broad set of features to present data in structures, such as tables and charts, to group data within these structures, to allow calculations, and to add formatting. This reporting platform also facilitates access to a variety of organizational data sources, such as online transaction processing (OLTP) systems or data warehouses that store relational or online analytical processing (OLAP) data. Reporting Services allows an enterprise report author to easily combine data from multiple sources into a single report. All types of structured data are supported for enterprise reports—relational, hierarchical, and multidimensional data. To access data sources not explicitly supported by Reporting Services, custom data processing extensions can be added, which means the possibilities are endless.

In addition, Reporting Services allows the report author to design a report with consideration for its purpose. An ad hoc report author can add interactive sorting and apply filtering to a report, while an enterprise report author has access to a rich feature set that enables the development of both static and interactive reports for the full spectrum of the user community. For example, static enterprise reports, such as print-ready invoices for mailing, or interactive online reports, such as the presentation of key performance indicators accessible from the corporate portal, can be quickly and easily developed. Interactive reports that factor in how users need to explore and analyze data can also be designed. Parameters, dynamic visibility, and actions can be used individually or in combination to affect both the information visible in an enterprise report and the information's appearance.

Management

The management stage begins when a report or ad hoc reporting model is published to a server. This stage continues with the organization of the report or model with other content on the server and the performance of other administrative tasks, such as setting report properties, managing report execution, and applying security. Either a report author or an administrator is responsible for publishing a report to a centrally managed server. When the report is on the server, a report execution schedule can be established to update the report regularly, such as every Monday morning. In addition, security is applied to the report so that only certain users, perhaps executive management, can view the report.

Reporting Services provides mechanisms to publish reports and models to a central server through the authoring tool or through management tools. Once a report is online, security can be implemented to control access. Further, the execution of reports is configurable so that reports can be produced on demand or on a scheduled basis. Reporting Services includes all this functionality using a server infrastructure that can exist on a single server or be distributed across many servers or incorporated into a Web farm.

Access and Delivery

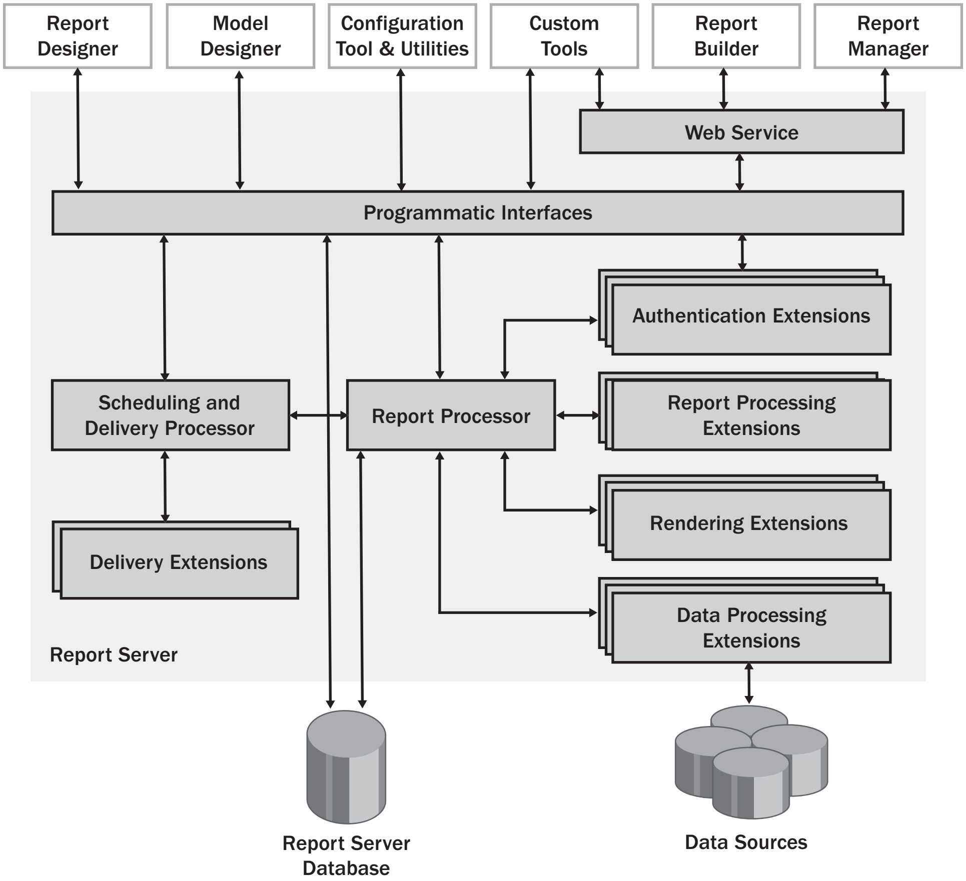
The access and delivery stage includes all activities related to the distribution of reports from a central repository, such as accessing reports online, rendering reports to various formats, saving and printing reports, and subscribing to reports. Some users, for instance, might choose to receive reports via e-mail as soon as these reports are executed each week or to receive a notification that a report is available for viewing online. Other users, by contrast, might prefer to view the reports online using the company intranet only as the need for information arises.

To support delivery, Reporting Services can produce reports using a variety of output formats which are referred to as rendering formats. A report can be made available through the intranet using a Web browser, or it can be sent embedded in an e-mail message or as an e-mail attachment in many formats, such as PDF or Excel. Reporting Services also provides flexible delivery mechanisms to support both push and pull distribution methods for internal and external users. Because Reporting Services is an extensible system, you can add rendering formats, different security frameworks, or alternative delivery options. In addition, the access or delivery of reports can be integrated into corporate applications.

Reporting Services Components

The requirements of a solid enterprise reporting solution are formidable, but Reporting Services meets this challenge with a set of integrated, multitiered components. Because Reporting Services is a Microsoft .NET-based platform that can use both a Web service and an application programming interface (API), it can be customized to fit within existing technical infrastructures. Furthermore, by separating components into discrete functional units, the Reporting Services architecture can be scaled to accommodate even the largest organization by distributing components across several servers. (You learn how to install these components in Chapter 2, "Installing Reporting Services.") Together, these components support the authoring, management, access, and delivery requirements of a reporting platform.

The Reporting Services architecture consists of three layers. The application layer includes two different client components for authoring—one to author enterprise reports, and the other to author ad hoc reports. The application layer also includes a client component for managing one or more Report Servers within a single server management interface, and a server component called Report Manager, which is installed on a Web server and used for report access and for some server management tasks. The server layer is the Report Server where all the processing and management of the reporting platform occurs. The data layer includes data providers to access data sources used in reports, as well as a pair of databases for the storage of reports and information used by Report Server. These components can be installed on a single server or distributed across several servers. The Reporting Services components are illustrated in the following figure:



Authoring Components

When Reporting Services is installed, a client component called Report Designer is added as a set of templates to the Microsoft Visual Studio 2005 development environment. If you don't have this version of Visual Studio installed, installation of Reporting Services will also install the requisite components so you can use Report Designer.

The easiest way to build enterprise reports is to use the Report Designer templates. If you're a report author, you don't need to have programming skills to effectively use this tool. However, if you're already an experienced programmer, you can also take advantage of the programmatic interfaces to build a custom authoring tool.

Report Builder is a thin client application that installs on your computer when you first launch this application through Report Manager or by accessing its URL on the Report Server. Report Builder is used for ad hoc reporting and, like Report Designer, does not require programming skills to create useful reports. In fact, you don't even need to know how to write a query to access data, since the query is written for you based on the items you select from the report model, which is created using Model Designer, another client component that is added to the Visual Studio environment.

Report Designer

As an enterprise report author, you can use a graphical interface to build feature-rich reports using drag-and-drop techniques to create a query to retrieve data and to define the layout and appearance of data in the report. You can use ActiveX Data Objects (ADO) .NET-managed data providers to access many OLE DB and Open Database Connectivity (ODBC) data sources. If you need access to other data sources, you can build your own data providers. After defining a query to retrieve data from a selected data source, you use Report Designer to place data into one or more structures. You also use Report Designer to apply calculations to the data, as well as to access a complete set of features that support presentation options such as formatting and visibility. The result is a report definition in the form of an XML document using a nonproprietary schema known as Report Definition Language (RDL). You learn about Report Designer and Report Definition Language in Chapter 4, "Developing Basic Reports."

Report Designer also includes a preview version of the functionality used by the server to produce reports, so you can test a report before putting it on the server. That way, you can get an idea of how the users will see the report while you're still developing it. When you're ready to publish the report to the server, you use the Visual Studio build and deploy processes.

Report Builder

If you're an ad hoc report author, you employ a separate graphical interface to build simple reports by using drag-and-drop techniques to retrieve data from a single source and to define the layout and appearance of data in the report. Instead of defining a data source and building a query, as you do with Report Designer, you select a report model, which contains the data connection, metadata, and data relationships needed to produce and execute a query. In the current release of Report Builder, a report model can only access relational data in a SQL Server database. If you use the Standard, Enterprise, or Developer Edition, you can also access multidimensional data in an Analysis Services database. After selecting a report model, you select a table, matrix, or chart template, which becomes the structure into which you place data fields defined by the report model. You can also use Report Builder to add your own calculations, sort or filter data, and apply formatting, such as font colors or styles. You can view the report within Report Builder and, if you want to share it with other users of Reporting Services, you can publish the report to the server. You learn about Report Builder in Chapter 15, "Creating Reports with Report Builder."

Model Designer

The report model on which Report Builder depends is created using the Model Designer interface. You begin development of the report model by specifying a data source and specifying which tables and their relationships to use from that data source. Then you use a wizard to generate a report model, which will become the business description of the underlying data. You can make changes to the names of objects generated in the model, rearrange and organize these objects to simplify the users' navigation in the model, and remove unneeded objects. You learn about Report Builder in Chapter 8, "Building Report Models."

Programmatic Interface for Authoring

Using the Reporting Services API, application developers can build custom applications to create reports or add functionality to reports. In Chapter 16, "Report Authoring with Custom Development," you learn how to use custom code to extend the authoring capabilities of Reporting Services and how to generate RDL files programmatically.

Management Components

After you've installed Reporting Services, you can use the Reporting Services Configuration tool to set or modify configuration settings. The Report Manager is installed on a Web server and used for both management and for access and delivery tasks. Report management capabilities are also accessible in Microsoft SQL Server Management Studio. All of the tasks you can perform using Report Manager can also be performed within this management component if it's installed on your local computer. Additionally, Reporting Services provides command-line utilities for specific server management tasks. You also have the option to build your own Windows or Web-based management tools using the Reporting Services API.

Reporting Services Configuration Tool

The Reporting Services Configuration tool allows you to configure local or remote instances of Reporting Services. This tool is a client application you can employ to manage virtual directories used by the report server and Report Manager, define Windows accounts to run the Web and Windows services, create the report server database, manage encryption keys to protect report server data, and configure Simple Mail Transfer Protocol (SMTP) settings. Typically, you configure these settings once after installation, but as your reporting environment changes, you can use this tool to update the report server configuration.

Report Manager

Report Manager is a Web-based content management tool included with Reporting Services. Reporting Services separates administrative tasks into two main groups: content management and system management. If you're an administrator responsible for content management, you can use Report Manager to manage how reports are organized on the Report Server and how users interact with those reports. Content management using Report Manager is covered in more detail in Chapter 9, "Managing Content." As an administrator responsible for server resources and performance, you can use Report Manager to configure execution options (described in Chapter 9), to set security (described in Chapter 10, "Managing Security"), and to manage subscription and delivery options (described in Chapter 14, "Managing Subscriptions").

Microsoft SQL Server Management Studio

Microsoft SQL Server Management Studio is a new workstation component provided with Microsoft SQL Server 2005 as a single environment from which to administer any of the server components of SQL Server. This environment does not replace Report Manager, nor does it provide additional features. Instead, it gives you another way to perform the same activities. For example, you can manage content, configure security, and set options for report execution or subscriptions from the same management interface that you use to manage SQL Server or Analysis Services if you're responsible for administering these other server types. This interface can also be useful if you manage multiple report servers.

Command-Line Utilities

Reporting Services includes command-line utilities that allow you to manage a Report Server locally or from a remote location. The following command-line utilities are provided by Reporting Services for server administrators:

- **rsconfig** A connection management utility that can change the connection used by Report Server to connect to the ReportServer database.
- **rs** A script host that you can use to execute Microsoft Visual Basic .NET scripts for management tasks, such as publishing reports or copying data between ReportServer databases.
- **rskeymgmt** An encryption key management tool that you can use to back up encryption keys for future recovery of a database or to change encrypted data used by a Report Server.

You learn more about these utilities in Chapter 11, "Managing Server Components," except for the *rs* utility, which is discussed in Chapter 17, "Programming Report Server Management."

Programmatic Interface for Management

You can also use the Reporting Services API to perform server management activities, such as publishing or deleting reports. You can build your own application, or you can build Visual Basic .NET scripts to use in conjunction with the *rs* utility to perform administrative tasks on Report Server. You learn how to use the Reporting Services API for server management in Chapter 17.

Access and Delivery Components

The components of Reporting Services that are involved in the access and delivery of reports break down into two groups: client components and server components. You can choose to use the supplied client components or build your own client applications. However, you must use the core server components of Reporting Services, although you can use custom or third-party applications to extend the capabilities of the server components.

Report Manager

Report Manager is not just a management tool. The user community also uses Report Manager to access reports and subscribe to reports. You learn about general report access using Report Manager in Chapter 12, "Accessing Reports."

Processor Components

Report Server is the heart of Reporting Services. Although administrators interact with Report Server using the management components, the bulk of activity that occurs on Report Server is related to supporting the access and delivery of reports. Report Server runs as a Web service, allowing Report Manager, your own custom programs, and third-party applications to access server processes.

Report Server uses Microsoft Internet Information Services (IIS) to receive requests, and then activates the applicable subcomponents in response to the requests. There are two processor subcomponents of Report Server that act as command central to manage these requests and the corresponding output returned by the other subcomponents. The core processor, Report Processor, handles all requests related to the execution of reports and the production of the final output. To complete these requests, the Report Processor calls other subcomponents, referred to as *extensions*, to handle data processing, rendering, and security. The Scheduling and Delivery Processor responds to scheduled events and delivers reports. This processor uses delivery extensions to send reports to their destinations.

Report Processor The Report Processor is responsible for retrieving the report layout from the report definition and merging it with data returned from the query included in the report definition. At this point, the report is in an *intermediate format*. What happens next depends on the report request. If a user wants to see the report online, the intermediate format is passed to the appropriate rendering extension so the proper output format can be created, such as a Web page.

On the other hand, if the request is to generate a *report snapshot* (which is a report at a specific point in time), the intermediate format is stored in the ReportServer database. When a user requests the snapshot, the Report Processor retrieves the intermediate format, calls the rendering extension, and then sends the final format of the report to the user.

The Report Processor also manages requests for report models. These requests occur when a user browses a report model or runs a report in Report Builder, including drillthrough reports.

By separating the presentation processing from data retrieval and rendering, multiple users can view the same report at the same time, and each can change the viewing format. You learn more about execution options in Chapter 9.

Scheduling and Delivery Processor As its name implies, the Scheduling and Delivery Processor is responsible for running scheduled reports and for delivering reports to a location or a device on a scheduled basis. It uses SQL Server Agent to process schedules. When the applicable time arrives, SQL Server Agent sends instructions related to the schedule to the Scheduling and Delivery Processor. The report is rendered and passed to the delivery extension to send the report to specified recipients or a target location on a file share.

A report snapshot is an example of a scheduled report. The schedule information is specified by a user or administrator using Report Manager and stored in the ReportServer database. When the Scheduling and Delivery Processor finishes processing the snapshot, the intermediate format is stored in the ReportServer database. When a user wants to view the report, Report Server takes over by retrieving the intermediate format of the snapshot and calling the specified rendering extension to finalize the output. You learn more about scheduling reports in Chapter 9.

When users subscribe to reports, the desired delivery extension is selected and the time of delivery is specified. When the time comes to deliver a report, the Scheduling and Delivery Processor gets a rendered report and then passes the report to the applicable delivery extension. Delivery options for subscriptions are described in Chapter 14.

Server Extensions

Server extensions are used to perform specific functions. Reporting Services uses five different types of server extensions: authentication, data processing, report processing, rendering, and delivery. Over time, you can expect to see more extensions available for Reporting Services, whether developed by Microsoft or by other commercial software developers.

Authentication Extensions Authentication extensions are used to define the authorization model used by Reporting Services. Only one authentication extension is supplied, which supports Microsoft Windows and SQL Server security. You can, of course, create your own authentication extension to integrate Reporting Services with another security architecture.

Data Processing Extensions Data processing extensions are responsible for processing the query requests received from the Report Processor. The query request includes a data source, a query, and possibly, query parameters. The applicable data processing extension then opens a connection to the data source, returns a list of field names from the query, executes the query, and retrieves the query results, which are then returned to the Report Processor.

Reporting Services comes with six data processing extensions: SQL Server, Analysis Services, Oracle, OLE DB, ODBC, and XML. However, you can also use any ADO.NET data provider or build your own data processing extensions. You learn how to create a custom data processing extension in Chapter 16.

Report Processing Extensions Report processing extensions are used to process custom report items that may be embedded in a report. The Report Server already knows how to process standard report items, such as tables and text boxes, but if you add a custom report item—like a special gauge control or an embedded map from Microsoft MapPoint—then you need to provide an extension to handle the custom processing required by the new report item.

Rendering Extensions Rendering extensions are called by the Report Processor to take the data that was received from the data processing extension and merge that data with the report definition. The result is a finished report in a format specific to the device that will receive the report.

At the time of this writing, Reporting Services has the following rendering extensions: HTML, MHTML (MIME Encapsulation HTML), Excel, Acrobat PDF, CSV (comma separated values), and XML. As with other extension types, you can develop your own rendering extension to produce other output formats.

Delivery Extensions Reporting Services currently includes the following three delivery extensions:

- The e-mail delivery extension allows Reporting Services to embed a report in an e-mail message or send the report as an attachment. Alternatively, an e-mail notification can be sent that includes a link to the report. If the delivery is an e-mail notification, it can also be sent without the link to a pager, cellular phone, or any device that can receive a simple message.
- The file share delivery extension can be used to store reports in a centrally accessible location independent of the ReportServer database or as part of a report archive strategy.
- A null delivery provider is available for data-driven subscriptions to periodically load reports into the cache in advance of user viewing. This option is useful for reports that take a long time to execute.

You can also develop your own delivery extension to expand the delivery functionality of Reporting Services.

ReportServer Databases

Reporting Services centralizes report storage in two SQL Server databases. The ReportServer database stores information used to manage reports and resources, along with the reports themselves. In addition, this database is the storage location for security settings, encrypted data, data related to schedules and delivery, and information about extensions. The ReportServerTempDB database stores temporary data used for caching purposes. More information about these databases can be found in Chapter 11.

Programmatic Interface for Access and Delivery

You can use the Reporting Services API to create assemblies when you need to accommodate specialized security, data processing, rendering, or delivery scenarios. In Chapter 16, you learn how to build a custom data processing extension. The Reporting Services API also enables you to develop your own applications to allow users to view reports or to produce reports using different formats. You learn more about custom reporting in Chapter 18, “Building Custom Reporting Tools.”

Chapter 1 Quick Reference

This term	Means this
Report	Information that is structured and formatted for print or online viewing.
Enterprise reporting	Sharing of information on a regular basis across a wide audience.
Ad hoc reporting	Reviewing, and possibly sharing, limited information with limited formatting requirements on an as-needed basis.
Enterprise reporting life cycle	The process of authoring, managing, and accessing reports.
Extensions	Subcomponents of Report Server used to provide specific functionality, such as data processing, rendering, report processing, authentication, and delivery.
Intermediate format	The result of merging data from a query with layout information from a report definition. The intermediate format is sent to a rendering extension to produce the final output, such as a Web page or an Excel file.
Report snapshot	A report that preserves a record of data at a point in time. A report snapshot is stored in the ReportServer database in its intermediate format and rendered only when a user requests the report.

Chapter 3

Building Your First Report

After completing this chapter, you will be able to:

- Use the Report Designer wizards to create a simple tabular report.
- Publish a report solution.
- Use Report Manager to manage report properties.
- Use the HTML Viewer to access and export a report.

In Chapter 1, "Understanding Reporting," you learned about the three stages of the reporting life cycle: authoring, managing, and accessing reports. In Chapter 2, "Installing Reporting Services," you learned how to install and configure Reporting Services, so you should be ready to go exploring now. In this chapter, rather than review each component of Reporting Services in detail, you take a tour of it. You visit each stage of the reporting cycle as you build, manage, and review your first report, and you also learn about the key components of Reporting Services.

You start your tour by authoring a simple report using wizards in the Report Designer, which will enable you to set up and design the report. You also use the Report Designer to polish and publish your report. Then, you move on to the management stage and use Report Manager to update the report's description and execution properties. Finally, you wrap up your tour in the access stage by using Report Manager to explore the report online and to export it as a Microsoft Excel file. When finished, you wind up with a high-level understanding of the various components of Reporting Services and the way they work together to create a powerful reporting platform.

Authoring a Report

The process of authoring, or building, a report consists of several steps. The first step is to define a Reporting Services *data source*, which packages information about where the data to be used in your report is stored. To create a data source, you need to know which server hosts the data and which database or file stores the data, as well as have the credentials with permission to retrieve that data. Each report that you author must have at least one data source defined. Data sources are covered in more detail in Chapter 4, "Developing Basic Reports."

The second step in building a report is to create a *dataset* for the report. An important component of the dataset is a query, which requires that you know the language and syntax used to retrieve data. For example, if your report will use data from a Microsoft SQL Server database, you'll need to be able to create a Transact-SQL query (or know someone who can write it for you!). A dataset also includes a pointer to the data source and other information that's used when the query executes. When you use the Report Server Project Wizard, as you will in this chapter, you can define only one dataset, but you'll learn how to work with multiple datasets in a single report in Chapter 7, "Building Advanced Reports."

The third and final step in the construction of your report is the creation of a *report layout,* which is the design template used by Reporting Services to arrange and format the data. The report layout includes the structure, or *data region*, into which data is placed when the report is processed, such as a table or matrix. You can set properties for each section of a data region to define style properties, such as font, color, and format. Additionally, you can set these properties for report items, such as the report title in a textbox or the report background, which gives you enormous flexibility to control the look and feel of your report.

In this chapter, you use the Report Server Project Wizard and the Report Wizard to help you start and build a new report. You'll learn another way to begin a report in Chapter 4. These wizards, which are provided within the Report Designer, are handy tools that walk you through the three main steps of authoring a report.

Starting a New Report

When you start a new report using the Report Server Project Wizard, you are creating Microsoft Visual Studio containers to hold your report, a project, and a solution. You must name these containers and provide a storage location for them on your computer's hard drive or on a network file share.

In this procedure, you'll create a new report project called Adventure Works and specify a storage location for the project.

Start the Report Server Project Wizard

1. Start SQL Server Business Intelligence Development Studio.

   Notice the title of the application is Microsoft Visual Studio. SQL Server Business Intelligence Development Studio *is* Visual Studio. You are simply using a shortcut from the Microsoft SQL Server 2005 program group to access Visual Studio.

2. On the File menu, point to New, and then click Project.

   The New Project dialog box appears. Templates are organized by Project Type, represented as folders in this dialog box.

> **Note** If this is your first time working with Visual Studio, you might not be familiar with the way that items are organized in this environment. A report is placed inside of a project, which you can think of as a folder that organizes many reports into a collection. Because you're using the Report Server Project Wizard, you can work with only one project right now. However, you'll be adding reports to this project as you progress through this book. When you publish all reports in a project, they are automatically organized into the same folder on the Report Server.

3. Click Report Server Project Wizard.
4. Type a name for the project: **Adventure Works**.

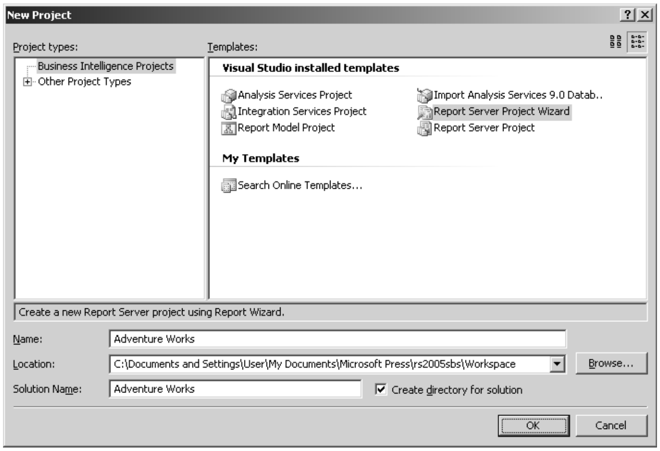
   Notice that as you type, the text in the Solution Name box of the New Project dialog box changes to match the project name. You have the option of changing the solution name later if you change your mind.

> **Note** In the same way that a project is a container for a report, a solution is a container for one or more projects. Visual Studio lets you work with only one solution at a time, but you can access any project within the open solution.

5. Type a location for the project: **C:\Documents and Settings\<username>\My Documents\Microsoft Press\rs2005sbs\Workspace**.

> **Important** Be sure to replace the placeholder <username> with the name you use to log into your computer.

The New Project dialog box looks like this:


New Project
Project types:
Business Intelligence Projects
Other Project Types
Templates:
Visual Studio installed templates
Analysis Services Project
Integration Services Project
Report Model Project
Import Analysis Services 9.0 Datab..
Report Server Project Wizard
Report Server Project
My Templates
Search Online Templates...
Browse...
OK
Cancel

Create a new Report Server project using Report Wizard.

Name: Adventure Works
Location: C:\Documents and Settings\User\My Documents\Microsoft Press\rs2005sbs\Workspace
Solution Name: Adventure Works
☑ Create directory for solution

6. Click OK to continue.

   The Welcome page of the Report Wizard is displayed. Note that this wizard is different from the Report Server Project Wizard. The Report Server Project Wizard lets you create a solution, a project, and a report in one step, and then launches the Report Wizard. You can use the Report Wizard any time you want to add a report to an existing project using a wizard interface. (This is explained in more detail in Chapter 4.)

If you want to bypass this page of the Report Wizard in the future, you can select the check box here to disable the Welcome Page.

7. Click Next.

Connecting to a Data Source

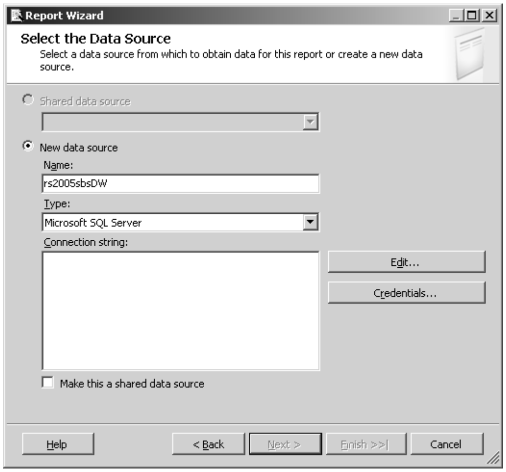
The next step of the Report Wizard allows you to specify connection information. Here you identify the server and database hosting the data. If necessary, you can also supply credentials information to be used by Reporting Services for authentication when querying the database.

In this procedure, you'll create a data source that defines a connection to the rs2005sbsDW database in your SQL Server using Microsoft Windows authentication.

Select a data source

1. Type a name for the data source: **rs2005sbsDW**.
2. Select Microsoft SQL Server in the Type drop-down list, if necessary.

   The Select The Data Source page of the Report Wizard now looks like this:



Report Wizard

Select the Data Source

Select a data source from which to obtain data for this report or create a new data source.

○ Shared data source

● New data source

Name: rs2005sbsDW

Type: Microsoft SQL Server

Connection string:

☐ Make this a shared data source

You can choose from seven connection types: Microsoft SQL Server, OLE DB, Microsoft SQL Server Analysis Services, Oracle, Open Database Connectivity (ODBC), XML, or Report Server Model. Once you select a connection type, you can type a connection string manually, or you can click Edit to use the Connection Properties dialog box to generate the connection string automatically. By default, the data source you create here will be available only to the current report, which allows you to manage its usage separately from other reports. You can select the Make This A Shared Data Source check box at the bottom of the dialog box to allow this data source to be shared with other reports, which simplifies the management of data sources in general.

3. Click Edit.

   The Connection Properties dialog box is displayed. Because you selected Microsoft SQL Server as the connection type on the Select The Data Source page of the Report Wizard, the data source defaults to Microsoft SQL Server (SqlClient).

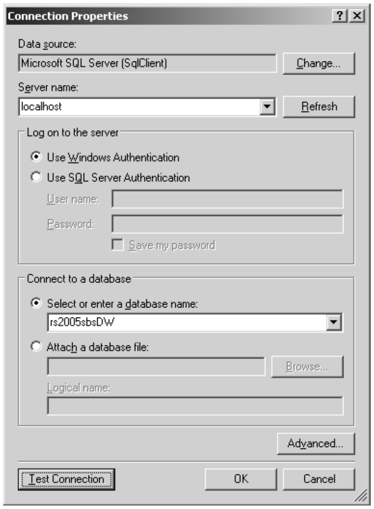
4. Type **localhost** or the name of the SQL Server instance that you are using in the Server Name box.

> **Note** This book assumes that you have all Reporting Services components and SQL Server installed on one computer. In a real-world environment, there are advantages to using localhost instead of a SQL Server instance since you can easily reuse the data source when moving from development to production if everything is similarly contained in a single machine. However, if you maintain separate instances of SQL Server, this strategy will not be useful.

5. Click Use Windows Authentication.

6. Select or enter rs2005sbsDW in the Select Or Enter A Database Name drop-down list.

   The Connection Properties dialog box looks like this:



Data source: Microsoft SQL Server (SqlClient)

Server name: localhost

Log on to the server

● Use Windows Authentication

○ Use SQL Server Authentication

Connect to a database

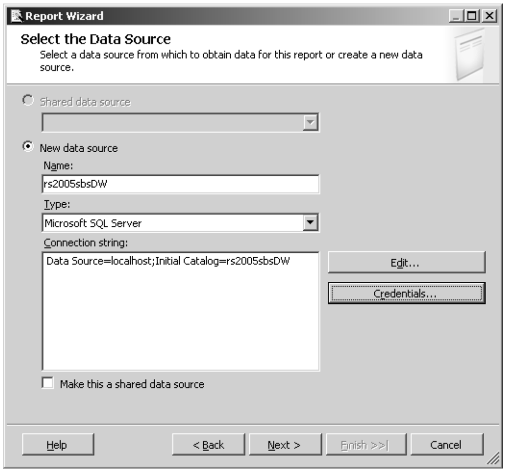
● Select or enter a database name: rs2005sbsDW

○ Attach a database file:

7. Click Test Connection to make sure you can connect to the rs2005sbsDW database, and then click OK to close the confirmation dialog box.

8. Click OK to close the Connection Properties dialog box.

The current page of the Report Wizard looks like this:



Report Wizard

Select the Data Source

Select a data source from which to obtain data for this report or create a new data source.

○ Shared data source

● New data source

Name: rs2005sbsDW

Type: Microsoft SQL Server

Connection string: Data Source=localhost;Initial Catalog=rs2005sbsDW

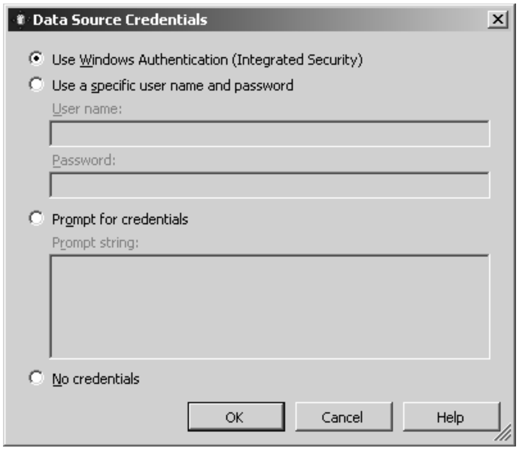
☐ Make this a shared data source

Notice the connection string generated for your SQL Server data source: `Data Source =localhost;Initial Catalog=rs2005sbsDW`. Remember that you can also type in a connection string for a data source, but it must use the syntax of the database to which Reporting Services will connect.

Now you have defined a data source that contains the information that Reporting Services needs to connect to the database it will use to retrieve data for your report. The data source includes a connection type, a connection string, and the credentials that will be used when the database is queried.

9. Click Credentials.

   The Data Source Credentials dialog box is displayed:



Data Source Credentials

● Use Windows Authentication (Integrated Security)

○ Use a specific user name and password

User name:

Password:

○ Prompt for credentials

Prompt string:

○ No credentials

You can click the applicable option to override the authentication method you specified in the Connection Properties dialog box. Authentication methods include Windows Authentication, a single user's credentials, a prompt at run time for the user's credentials, or no credentials at all. (You'll learn more about credential management in Chapter 9, "Managing Content.")

10. Click Cancel.
11. Click Next.

Getting Data for the Report

In this next step of the Report Wizard, you design the query that will be displayed in the report. The query must conform to the relational database syntax you defined in the data source. You must get this query correct, or you won't be able to continue with this wizard.

In this procedure, you'll paste in a query that summarizes the Adventure Works sales for each employee by year, sales territory group, and sales territory country.

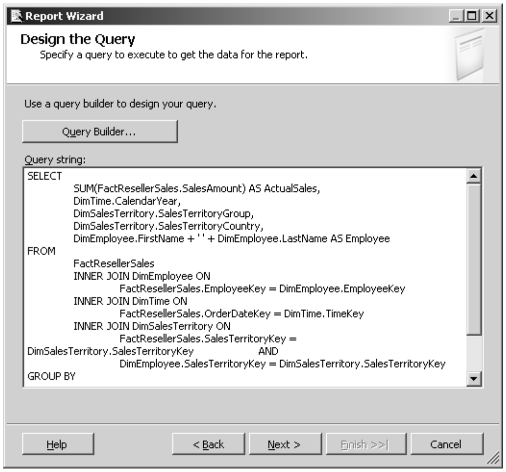
Design a query

1. Start Microsoft Notepad.
2. On the File menu, click Open.
3. Open the Sales Summary.txt file in the C:\Documents and Settings\<username>\My Documents\Microsoft Press\rs2005sbs\chap03 folder.
4. Copy the following query entirely:

```
SELECT
   SUM(FactResellerSales.SalesAmount) AS ActualSales,
   DimTime.CalendarYear,
   DimSalesTerritory.SalesTerritoryGroup,
   DimSalesTerritory.SalesTerritoryCountry,
   DimEmployee.FirstName + ' ' + DimEmployee.LastName AS Employee
FROM
   FactResellerSales
   INNER JOIN DimEmployee ON
         FactResellerSales.EmployeeKey = DimEmployee.EmployeeKey
   INNER JOIN DimTime ON
         FactResellerSales.OrderDateKey = DimTime.TimeKey
   INNER JOIN DimSalesTerritory ON
     FactResellerSales.SalesTerritoryKey = DimSalesTerritory.SalesTerritoryKey
AND
         DimEmployee.SalesTerritoryKey = DimSalesTerritory.SalesTerritoryKey
GROUP BY
   DimTime.CalendarYear,
   DimSalesTerritory.SalesTerritoryGroup,
   DimSalesTerritory.SalesTerritoryCountry,
   DimEmployee.FirstName + ' ' + DimEmployee.LastName
```

5. Paste the copied query into the Query String box on the Design The Query page of the Report Wizard.

   Now the current page of the Report Wizard looks like this:



Design the Query

Specify a query to execute to get the data for the report.

Use a query builder to design your query.

```
SELECT
        SUM(FactResellerSales.SalesAmount) AS ActualSales,
        DimTime.CalendarYear,
        DimSalesTerritory.SalesTerritoryGroup,
        DimSalesTerritory.SalesTerritoryCountry,
        DimEmployee.FirstName + ' ' + DimEmployee.LastName AS Employee
FROM
        FactResellerSales
        INNER JOIN DimEmployee ON
                FactResellerSales.EmployeeKey = DimEmployee.EmployeeKey
        INNER JOIN DimTime ON
                FactResellerSales.OrderDateKey = DimTime.TimeKey
        INNER JOIN DimSalesTerritory ON
                FactResellerSales.SalesTerritoryKey =
DimSalesTerritory.SalesTerritoryKey             AND
                DimEmployee.SalesTerritoryKey = DimSalesTerritory.SalesTerritoryKey
GROUP BY
```

> **Note** Instead of typing or pasting in a query string, you can also click Query Builder to open the Query Builder to create a SELECT statement using a graphical interface. If you've used the Query Builder in SQL Server 2000 Enterprise Manager or the Query Designer in Microsoft SQL Server Management Studio for SQL Server 2005, you'll be in familiar territory. If you haven't used either of these applications, you can learn more about the Query Builder in Chapter 7.

   This query will be used to retrieve data from the defined data source for use in your report. The format of the query depends on the data source you selected. For this procedure, because you selected a Microsoft SQL Server data source, you use Transact-SQL to build your query.

   The query that you create is just one of several items stored in a dataset. As you learned earlier in this chapter, a dataset is a container for a pointer to the data source and the query you design. (You'll learn more about designing queries to create a dataset in Chapter 4.) In general, you can type a query directly into the Query String box, use the Query Builder button to open the Query Builder, or paste in a query that has been tested first in Query Analyzer or saved in a file.

6. Click Next.

   When you click Next, the query is validated against the data source. If there is any problem, such as an invalid column name, an error message will be displayed in the bottom section of the Design the Query page. You will not be able to continue past this page of the wizard until you correct the error.

Structuring Data in the Report

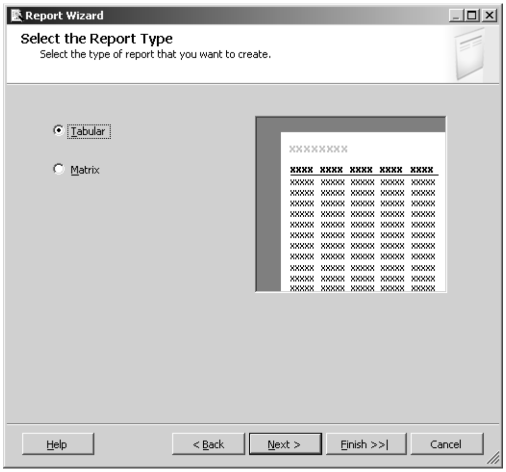
After defining the data source and the dataset, you're ready to move on to design considerations. Now you select a report type that defines how the data is structured in the report. In the wizard, you can choose between a tabular or a matrix report type only. You also arrange the data within the selected structure and finish the design by applying a style template. These steps make it easy to create a nice-looking report without a lot of effort, but you'll still have an opportunity to make adjustments to the layout and style before you publish the report.

In this procedure, you'll select the tabular report type for your report.

Choose a report type

Click Tabular.

1. The Report Wizard page looks like this:



Select the Report Type

Select the type of report that you want to create.

● Tabular

○ Matrix

Notice that the Finish button is now enabled. You have, at this point, created a basic report that is ready for publishing. Now you can decide how you want to proceed. You could click Finish and make any desired modifications using the Report Designer. However, to find out everything you can do with this tool, you'll continue designing your report with the wizard.

The *report type* defines the structure, or data region, of the data that is returned by the query you design. The Report Wizard allows you to present this information as either a table or a matrix. (You do have more options, but the Report Wizard limits you to these two data regions, referred to as report types in the wizard.) The main difference between these two types of data regions is the number of columns. A table has a fixed number of columns, whereas a matrix has a variable number of columns that is determined by the query results. You'll find more information about these and other data regions in Chapter 6, "Organizing Data in Reports."

2. Click Next.

Placing Data in the Report Structure

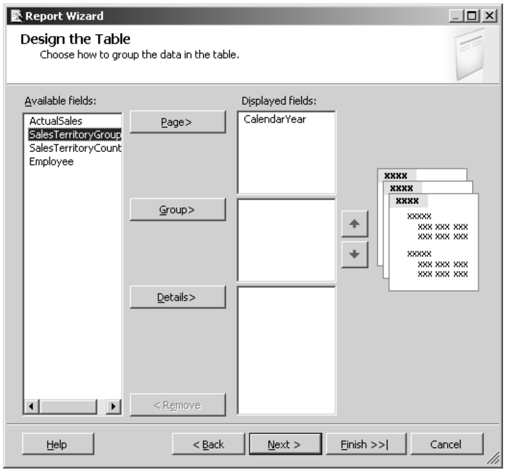
In this step of the Report Wizard, you arrange the data within the report type that you selected. This process determines how data is grouped and the order in which it is displayed. You can think of grouped data as the vertical sections of a report (although groups can be displayed next to each other), and the data order as the sequence in which the data is presented in the same row—vertically for groups and horizontally for columns.

In this procedure, you'll arrange the five fields produced by the query to build a report that displays the ActualSales amount for each Employee as detail rows, in groups by SalesTerritoryGroup and SalesTerritoryCountry, with a page break for each CalendarYear.

Arrange data on the report

1. Click CalendarYear, and then click Page to place the *CalendarYear* field in the Page section of the Displayed Fields list.

   The Report Wizard looks like this:



When you place a field in a display section, the corresponding section in the sample table is highlighted to show you where the field will appear in your report. Each column of data returned by the query is linked, or mapped, to a report field that is displayed in the Available Fields list until assigned to a section of the data region. When assigned to a data region's section, the report fields appear in the Displayed Fields list. The section to which you assign the field determines whether you see detail rows, aggregated rows, or both types of rows in the report. Assignment of fields to data regions and the use of aggregations are discussed more thoroughly in Chapter 4 and Chapter 5, "Working with Expressions."

Because you are using a tabular report type, you can assign fields to the page, group, or details section of the report. For example, a field assigned to the page section will not be included in the table in the report, but will instead be placed in a textbox positioned at the top-left corner of the report. Each distinct value for a page field creates a page break in the report. Fields added to the group section of the report are used to break the table into separate sections, which can include subtotals by section.

The table rows are built from the values for the fields assigned to the details section of the report. There is one table row for each row returned by the defined query. A numeric field in the details section is summed up into the subtotals if you select the option to include subtotals. You can decide later whether you want to hide the details in the report if you prefer to display just summary information.

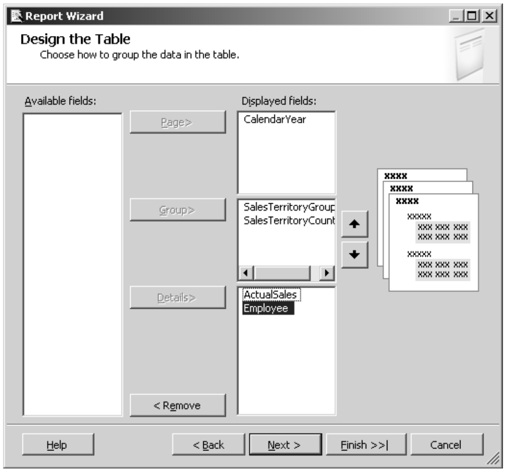
> **Note** If you choose a matrix report type, the field assignment is slightly different. The wizard still includes the page and details sections, but the group section is replaced by sections for columns and rows. You'll need to assign at least one field to each of these sections to build a matrix, which is also known as a crosstab. Matrix reports are covered in Chapter 6.

2. Click SalesTerritoryGroup, and then click Group to place the field in the Group section. Repeat for SalesTerritoryCountry. Alternatively, you can use the drag-and-drop feature to move a field from the list of available fields to the appropriate section.

   The order in which you add fields to each section determines the sequence in which the data is displayed in the report. The fields in the Group section will be displayed in order from top to bottom or from left to right, depending on the style template that you select in a later step in the wizard. Fields in the Details section will be displayed in columns in order from left to right.

3. Drag ActualSales to the Details section, which is the bottom section of the Displayed Fields list, and then drag Employee to the same section.

   The Design The Table page of the Report Wizard looks like this:



Report Wizard

Design the Table

Choose how to group the data in the table.

Available fields:

Displayed fields:

CalendarYear

SalesTerritoryGroup
SalesTerritoryCount

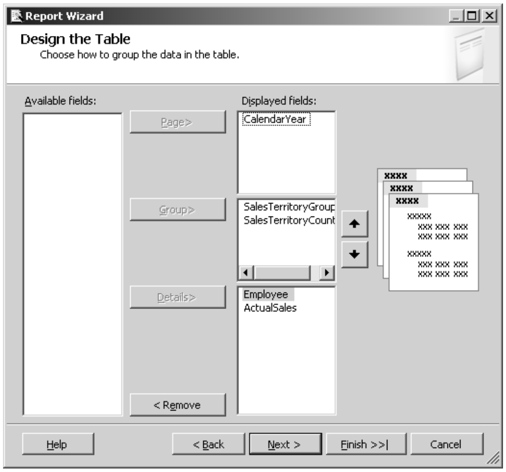
ActualSales
Employee

Tip Even after the fields are placed into the Displayed Fields list, you can still rearrange them to affect their order in the respective sections.

If you had selected the Matrix option on the previous page, Select the Report Type, you would see the Design the Matrix page here instead of the Design The Table page.

4. Click Employee, and then click the Up button to move Employee above ActualSales. You can also drag and drop to rearrange fields within a data region.

   Now the page looks like this:



Report Wizard

Design the Table

Choose how to group the data in the table.

Available fields:

Displayed fields:

CalendarYear

SalesTerritoryGroup
SalesTerritoryCount

Employee
ActualSales

5. Click Next.

Applying a Style Template

In this step of the Report Wizard, you make your last design decision for your report. When you apply a style template, you define the look and feel of the report.

In this procedure, you'll define a block layout for the table, which includes group subtotals, and select the Bold style template.

Select a report style

1. Select Block.

Note This page of the wizard will not be displayed if you selected a matrix report type.

Notice that the sample layout changes to give you a preview of the block layout. Here, you are choosing a layout style for the tabular report that controls the placement of detail rows relative to aggregated rows on the report. You can also choose to include subtotals or enable drilldown. The difference between the layout options will become clearer when you can actually view your report. At that time, you'll see examples of the other layout styles for comparison.

On this page of the Report Wizard, you must select either a stepped or block report layout style. In a stepped layout, each distinct group value is arranged on its own row and in its own column. The drilldown option (which displays hidden detail data) is available only for stepped layout. By contrast, the block layout is more compact—you place data in each column and start a new row only for additional detail rows within the same group or for a new group value.

2. Select the Include Subtotals check box to include subtotals in your report.
3. Click Next.
4. Click each table style to preview the style in the Choose The Table Style page.

   The assignment of a style template to a tabular or matrix report sets the overall color theme and font usage for the report.
5. Click Bold to set the style for your report.
6. Click Next.

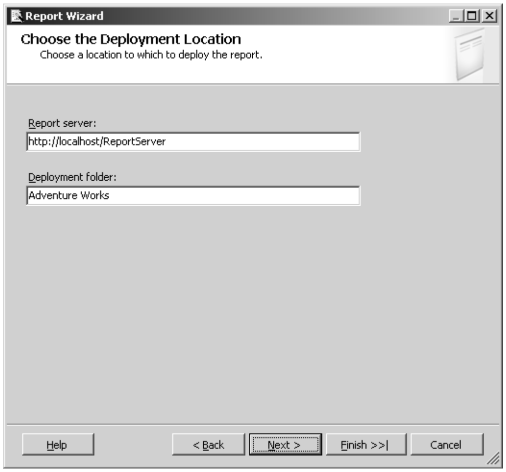
Finishing the Report Wizard

You're almost finished building your report by using the wizard. You've defined where to find the data, what data to include in the report, and how the data will look in the report. All that remains is to specify a location on the Report Server that will be the ultimate destination of your report when it is published and to give your report a name. You also have an opportunity to review a summary of the selections that you made throughout the wizard and to proceed to a preview of your report.

In this procedure, you'll provide the URL for your local Report Server, specify the Adventure Works folder for deployment, and name your report. When you're finished with the wizard, you will be able to preview the report.

Set report and project properties

1. Confirm that the current page of the Report Wizard looks like this:



Report Wizard

Choose the Deployment Location

Choose a location to which to deploy the report.

Report server: http://localhost/ReportServer

Deployment folder: Adventure Works

The deployment location is a URL for the Report Server that will host the report as well as the folder into which the report will be placed on the server. This step of the Report Wizard simply sets the project properties and does not actually *deploy*, or publish, the report to the Report Server. Notice that the default folder has the same name as your project. If this folder does not already exist, the folder will be created when you deploy the report. Otherwise, the report will be deployed to the existing folder.

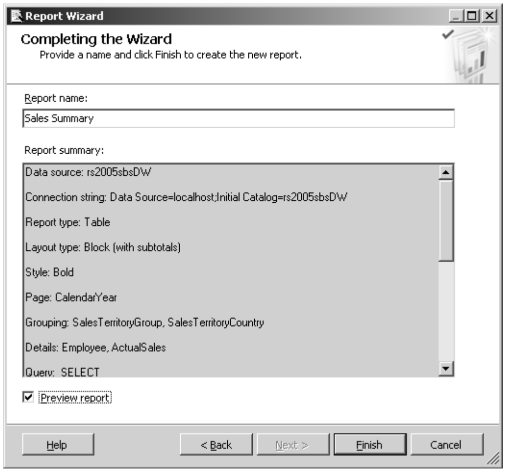
> **Important** The server name will not be validated in this step. If you enter an incorrect server name, deployment fails. You can update the project properties in the Solution Explorer if this occurs. You'll learn how to do this in Chapter 9.

2. Click Next.
3. Type **Sales Summary** in the Report Name box.

> **Important** If you use the name of a report that has already been deployed to the Report Server, you will overwrite the published report during deployment of the report in Visual Studio—but only if you deploy the report to the same folder as the existing folder. There will be no warning message during deployment that you are about to overwrite an existing report, so be careful when assigning names and folder locations to reports.

4. Scroll through the information in the Report Summary box to review your selections.
5. Select the Preview Report check box to preview your report.

The final page of the Report Wizard now looks like this:



Completing the Wizard

Provide a name and click Finish to create the new report.

Report name: Sales Summary

Report summary:

Data source: rs2005sbsDW

Connection string: Data Source=localhost;Initial Catalog=rs2005sbsDW

Report type: Table

Layout type: Block (with subtotals)

Style: Bold

Page: CalendarYear

Grouping: SalesTerritoryGroup, SalesTerritoryCountry

Details: Employee, ActualSales

Query: SELECT

☑ Preview report

At the completion of the Report Wizard, you can immediately preview the report. Sometimes you might prefer to make some additional changes to the report before you display the preview. If you do not select the Preview Report check box, the Report Designer displays your report in layout mode. If the report is in layout mode, you can easily switch to preview mode by clicking the Preview tab in the Report Designer.

Tip An important reason to preview the report is to check the size of the columns. The columns will all default to the same size and will probably not be wide enough for data. Also, you might need to adjust formatting for numeric values. You can fix these problems using the layout mode, and then review the fixes using the preview mode.

6. Click Finish.

Checking the Report Layout

When you finish designing your report, you need to preview it to check the layout and make some corrections to improve its appearance. You also need to make sure you get the data you expected and that the data is formatted correctly. When you finish making any necessary corrections, you wrap up this authoring stage of the reporting life cycle by publishing the report to the Report Server, where it can be accessed by the users.

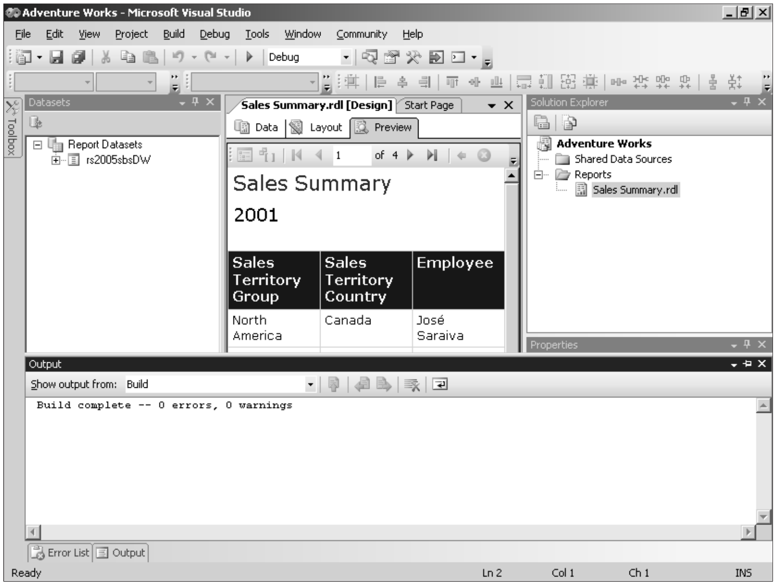
In this procedure, you'll explore your report in preview mode so you can see the results of the selections you made in the Report Wizard.

Preview a report

1. If you selected the check box to preview the report on the last page of the Report Wizard, you will see your report in preview mode. If not, just click the Preview tab in the Report Designer.

   When you display a report in preview mode, the query is executed and the query results are stored in a dataset and assigned to fields. The report is then rendered according to the assignment of the fields to data regions that you specified as well as the layout and style that you selected. At this point, the report format and the report data are merged to produce the preview that you can see in the Report Designer. In preview mode, you can interact with the report just as if it were published to the server so you can test the results before making it available on the Report Server.

   Your screen now looks like this:



Sales Summary

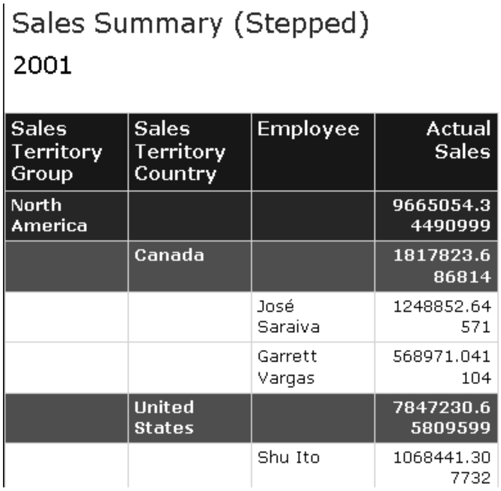
2001

Sales Territory Group	Sales Territory Country	Employee
North America	Canada	José Saraiva

```
Build complete -- 0 errors, 0 warnings
```

   Take a moment to review the layout of the data in the report. The *CalendarYear* field is displayed in the top-left corner. Just below the CalendarYear, you can see the column names in the table header with the details displayed in rows by groups. The first group is SalesTerritoryGroup, and the second group is SalesTerritoryCountry. Because these fields are defined as groups, their values are displayed only in the first row of details within that group. In the example, you can see actual sales amounts for employees in Canada which is grouped in North America. The row beneath these details is the group subtotal for Canada.

   Once you've closed the wizard, you can't return to it to make layout changes. Instead, you must create a new report, which you may find easier to do than making changes to the layout directly in the Report Designer. If you were to use the wizard to create a stepped report with subtotals (on the Choose The Table Layout page) using the same query, then the report would look like this:



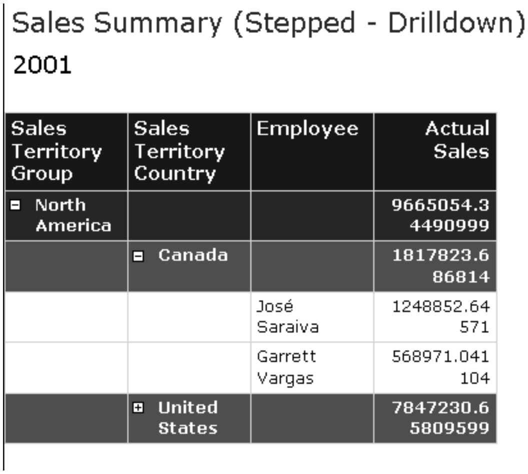
Sales Summary (Stepped)

2001

Sales Territory Group	Sales Territory Country	Employee	Actual Sales
North America			9665054.34490999
	Canada		1817823.686814
		José Saraiva	1248852.64571
		Garrett Vargas	568971.041104
	United States		7847230.65809599
		Shu Ito	1068441.307732

Notice that North America is now in its own row, and its subtotal is included on the same row. Then Canada appears by itself on the next row, followed by the detail rows. This report style is longer than the block layout.

If you had instead used the wizard to create a report with a stepped layout with drill-down selected, the report—with North America and Canada expanded to show the detail rows—would look like this:



Sales Summary (Stepped - Drilldown)

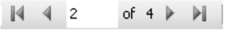
2001

Sales Territory Group	Sales Territory Country	Employee	Actual Sales
⊟ North America			9665054.34490999
	⊟ Canada		1817823.686814
		José Saraiva	1248852.64571
		Garrett Vargas	568971.041104
	⊞ United States		7847230.65809599

With drilldown, the user can click the plus sign to expand the report and click the minus sign to collapse the report at will. By default, the report is completely collapsed when it is opened.

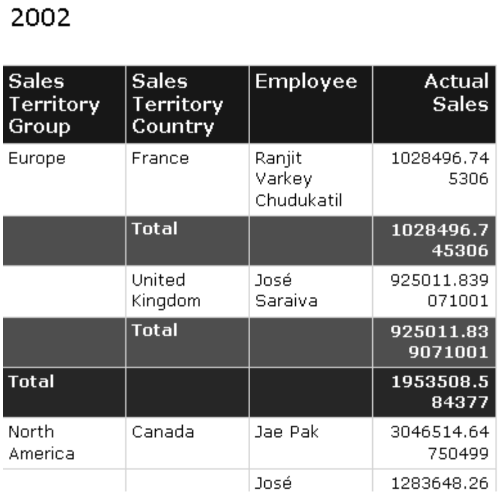
2. Scroll down to the bottom of the first page. Notice the group subtotal for United States. Beneath this group subtotal, you can see the group subtotal for SalesTerritoryGroup, which is the subtotal for Canada and the United States.
3. Click the Next Page button on the Preview toolbar to view the page for 2002.

   Since you assigned CalendarYear to the Page data region of the report, each page contains data for a separate year. You can use the page buttons on the Preview toolbar to navigate between pages, or you can type in the page number that you want to view as shown here:

4. Scroll to the bottom of the second page to see the layout when there are multiple values for SalesTerritoryGroup, then scroll back to the top of the page.
5. If you can't see the full width of the report, scroll horizontally to see the Actual Sales column.

   The top of your report looks like this:



2002

Sales Territory Group	Sales Territory Country	Employee	Actual Sales
Europe	France	Ranjit Varkey Chudukatil	1028496.745306
	Total		**1028496.745306**
	United Kingdom	José Saraiva	925011.839071001
	Total		**925011.839071001**
Total			**1953508.584377**
North America	Canada	Jae Pak	3046514.64750499
		José	1283648.26

Notice that the text is wrapping in several columns: SalesTerritoryCountry, Employee, and ActualSales. If you are trying to fit many columns onto a printed page, you may need to use text wrapping to fit a table within the available horizontal space. However, for online viewing, you generally have more room available and you can minimize vertical scrolling for the user if you widen each column to accommodate the maximum expected string length. In addition, the format of the ActualSales values can be improved.

Correcting Report Layout Issues

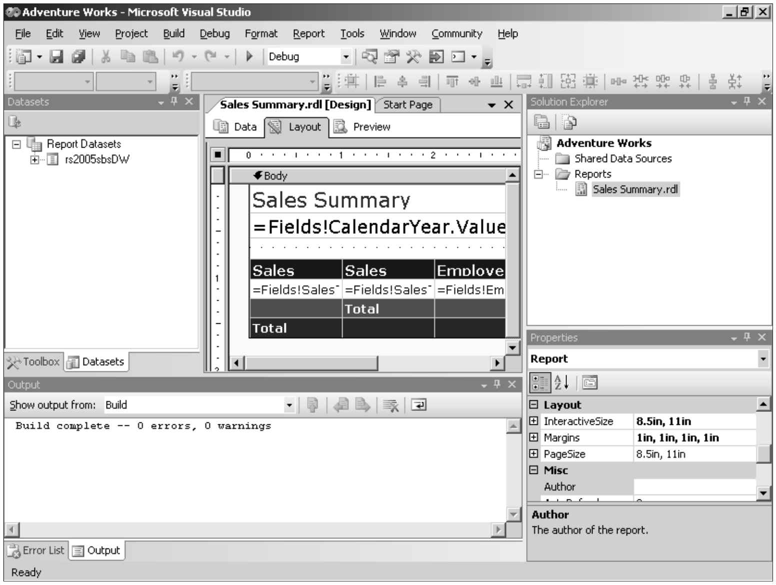
Preview mode in Report Designer allows you to see where you need to clean up your report, but you need to switch to layout mode in order to fix the problems. In layout mode, you can adjust every property of every element in the report, giving you complete control over everything that you can see. You can easily switch back and forth to test the results of your changes to the report in layout mode.

In this procedure, you'll use layout mode to improve the appearance of the report and check the results by again previewing the report.

Fix column sizes and data formatting in the report layout

1. Click the Layout tab.

   The report is displayed in layout mode:



Show output from: Build

```
Build complete -- 0 errors, 0 warnings
```

Properties

Report

Layout

InteractiveSize 8.5in, 11in

Margins 1in, 1in, 1in, 1in

PageSize 8.5in, 11in

Misc

Author

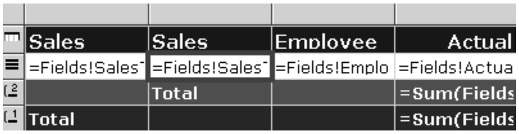
Author

The author of the report.

   Notice the rulers that appear both above and to the left of the report layout. You can use these rulers as a visual guide when making changes to the report, such as when resizing report items or positioning new report items.

2. Click any cell in the table to display the column and row handles.

   The table now looks like this:

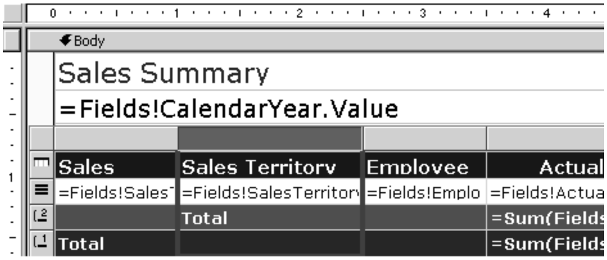


Sales	Sales	Emplovee	Actual
=Fields!Sales	=Fields!Sales	=Fields!Emplo	=Fields!Actua
	Total		=Sum(Fields
Total			=Sum(Fields

The column handles are the shaded cells that appear above the table, and the row handles are the shaded cells with icons that are shown to the left of the table. You use these handles to modify the table properties.

3. Position your cursor between the second and third column handles, and then click and drag to widen the second column, Sales Territory Country, to approximately 1.5 inches.

   Now the table looks like this:



Sales Summary

=Fields!CalendarYear.Value

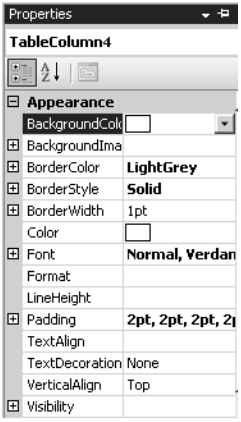
Sales	Sales Territory	Employee	Actual
=Fields!Sales	=Fields!SalesTerritor	=Fields!Emplo	=Fields!Actua
	Total		=Sum(Fields
Total			=Sum(Fields

   You can drag the column only when the cursor is properly positioned and the cursor changes to a double-headed arrow. Making the column bigger eliminates the text wrapping problem, but it also requires you to have some idea of the maximum length of the data that could appear in that column.

4. Position the cursor between the third and fourth column handles, and then click and drag to widen the Employee column to approximately 1.75 inches.

5. Right-click the fourth column handle, above Actual Sales, to select the entire column, and then click Properties.

   The Properties window for the selected column, named TableColumn4, is displayed in Visual Studio:



TableColumn4

Appearance

BackgroundCol	
BackgroundIma	
BorderColor	**LightGrey**
BorderStyle	**Solid**
BorderWidth	1pt
Color	
Font	**Normal, Verdan**
Format	
LineHeight	
Padding	**2pt, 2pt, 2pt, 2**
TextAlign	
TextDecoration	None
VerticalAlign	Top
Visibility	

6. Scroll through the Properties window, if necessary, to find the *Format* property, and then type **C0** in the *Format* property field to format the field as currency with no decimal places.

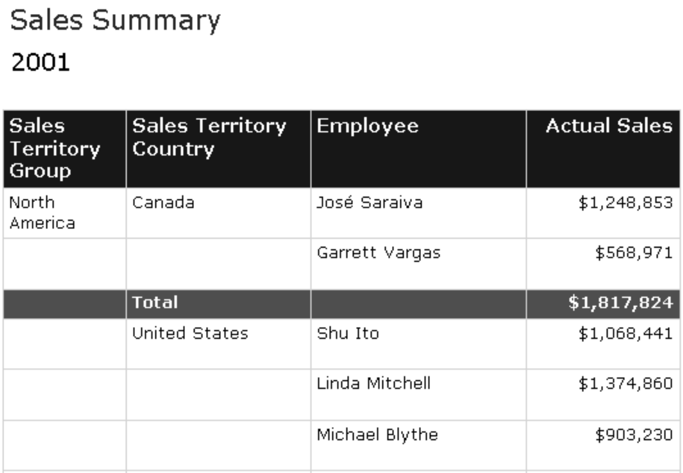
> **Note** Use .NET formatting strings to control the data display. You can find more information about formatting numeric strings online at *http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpguide/html/cpconstandardnumericformatstrings.asp*. Information about formatting date strings is located at *http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpguide/html/cpconDateTimeFormatStrings.asp*.

Scroll to the bottom of the Properties window to find the *Width* property, and then type **1.25in** to resize this column.

You can provide a specific measurement for the *Width* property when you require more granular control over the size of a column.

Click the Preview tab to preview the modified report.

The newly formatted report is displayed:



Sales Summary

2001

Sales Territory Group	Sales Territory Country	Employee	Actual Sales
North America	Canada	José Saraiva	$1,248,853
		Garrett Vargas	$568,971
	Total		$1,817,824
	United States	Shu Ito	$1,068,441
		Linda Mitchell	$1,374,860
		Michael Blythe	$903,230

The text wrapping problem is solved, and the format of the Actual Sales column is improved. Your first report is ready for publishing!

Publishing a Report

Now you'll wrap up the authoring stage of the reporting life cycle by publishing the report to the Report Server, where it can be accessed by the user community.

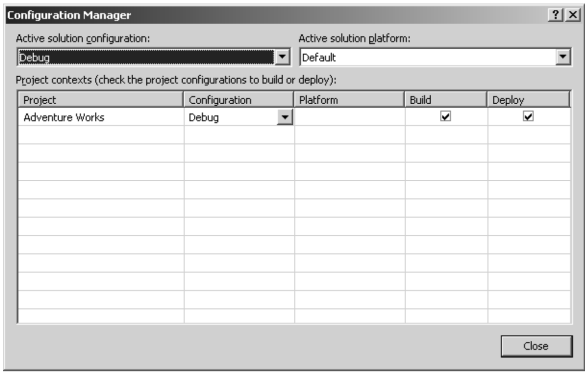
In this procedure, you'll deploy a report solution that enables you to publish your report to your local Report Server.

Deploy a report solution

1. On the File menu, click Save All.
2. In the Solution Explorer window, right-click the Adventure Works project at the top of the tree, and then click Properties.

   The Adventure Works Property Pages dialog box is displayed. You can see the Target ReportFolder and TargetServerURL properties for which values were provided on the Choose The Deployment Location page of the Report Wizard.
3. Click Configuration Manager.
4. Verify that the Deploy check box is selected.

   The Configuration Manager dialog box looks like this:



Active solution configuration: Debug

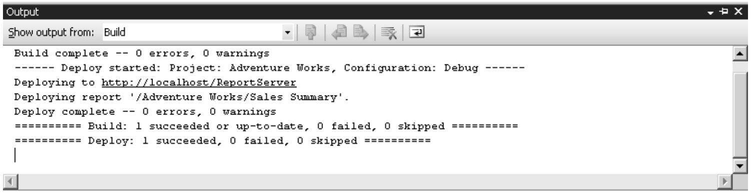
Active solution platform: Default

Project contexts (check the project configurations to build or deploy):

Project	Configuration	Platform	Build	Deploy
Adventure Works	Debug		☑	☑

5. Click Close, and then click OK.
6. On the Build menu, click Deploy Adventure Works.

   The Output window displays the progress of deployment. Deployment of the solution is complete when you see messages in the Output window announcing that the build and deploy operations succeeded:



```
Build complete -- 0 errors, 0 warnings
------ Deploy started: Project: Adventure Works, Configuration: Debug ------
Deploying to http://localhost/ReportServer
Deploying report '/Adventure Works/Sales Summary'.
Deploy complete -- 0 errors, 0 warnings
========== Build: 1 succeeded or up-to-date, 0 failed, 0 skipped ==========
========== Deploy: 1 succeeded, 0 failed, 0 skipped ==========
```

Instead of deploying an entire solution, you also have the option to deploy a single report or multiple reports within a project or solution. A report is published by using one of these deployment options to transfer it from Visual Studio to the Report Server. You can alternatively publish a report programmatically using a script or manually using the Web application called Report Manager, which you'll learn about in Chapter 9.

Managing a Report

You can manage published reports by using Report Manager, which is supplied by Reporting Services. Management of reports includes such activities as setting report properties and execution properties, managing content in folders, and applying security on the Report Server to control how users access and interact with reports. You perform only a few management tasks in this chapter. (You'll learn about all the management tasks in Part III, "Managing the Report Server.")

Reviewing Report Properties

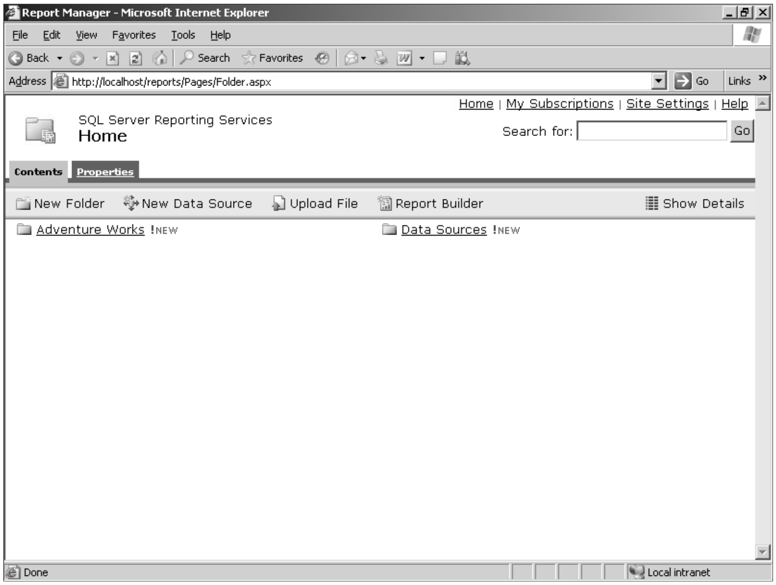
Each report has a set of properties pages that you must manage. You need to know how to use the Report Manager to find these properties and to review the types of properties you can manage.

In this procedure, you'll navigate from the Home page of Report Manager to the Properties page of your report.

Open the report's Properties page

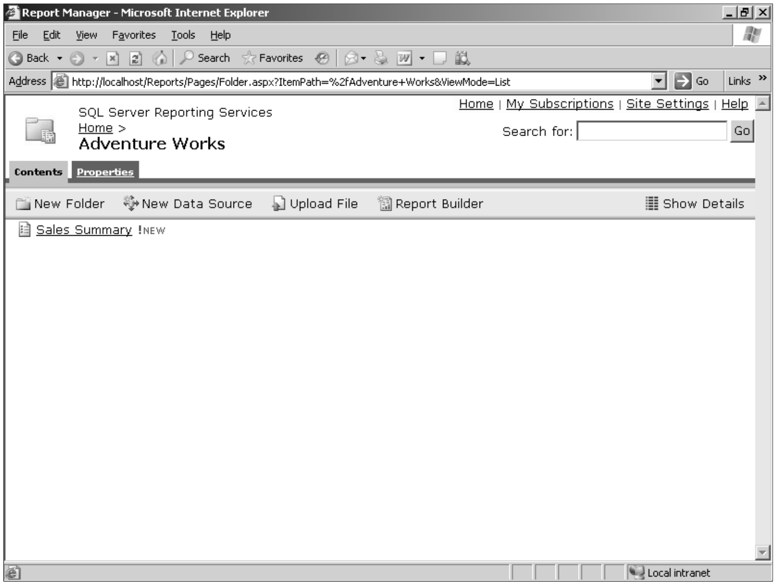
1. Open Internet Explorer.
2. Type the URL ***http://localhost/Reports*** to open the Report Manager.

   The Home page of Report Manager is displayed:


Report Manager - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost/reports/Pages/Folder.aspx
Go Links
Home | My Subscriptions | Site Settings | Help
SQL Server Reporting Services
Home
Search for: Go
Contents Properties
New Folder New Data Source Upload File Report Builder Show Details
Adventure Works !NEW
Data Sources !NEW
Done
Local intranet

3. Click the Adventure Works folder link.

The folder contents are displayed:

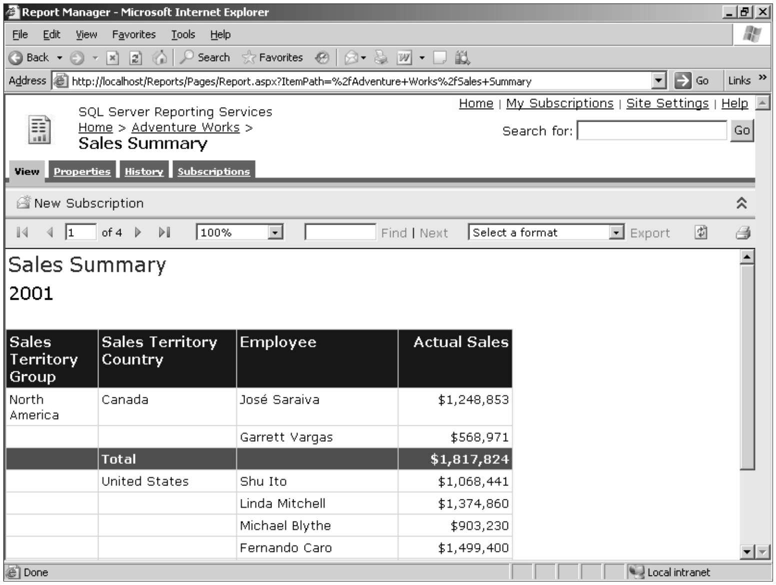

Report Manager - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address
Go Links
Home | My Subscriptions | Site Settings | Help
SQL Server Reporting Services
Home >
Adventure Works
Search for: Go
Contents Properties
New Folder New Data Source Upload File Report Builder Show Details
Sales Summary !NEW
Local intranet

Address: http://localhost/Reports/Pages/Folder.aspx?ItemPath=%2fAdventure+Works&ViewMode=List

Notice that this page has a Properties tab in addition to the Contents tab. (You'll learn more about managing folder properties in Chapter 9.) Currently, the Adventure Works folder contains only one report, the Sales Summary report that you just published.

4. Click the Sales Summary link.

   Reporting Services generates and displays the Sales Summary report:


Report Manager - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address
Go Links
Home | My Subscriptions | Site Settings | Help
SQL Server Reporting Services
Home > Adventure Works >
Sales Summary
Search for: Go
View Properties History Subscriptions
New Subscription
1 of 4
100%
Find | Next
Select a format
Export
Done
Local intranet

Address: http://localhost/Reports/Pages/Report.aspx?ItemPath=%2fAdventure+Works%2fSales+Summary

Sales Summary
2001

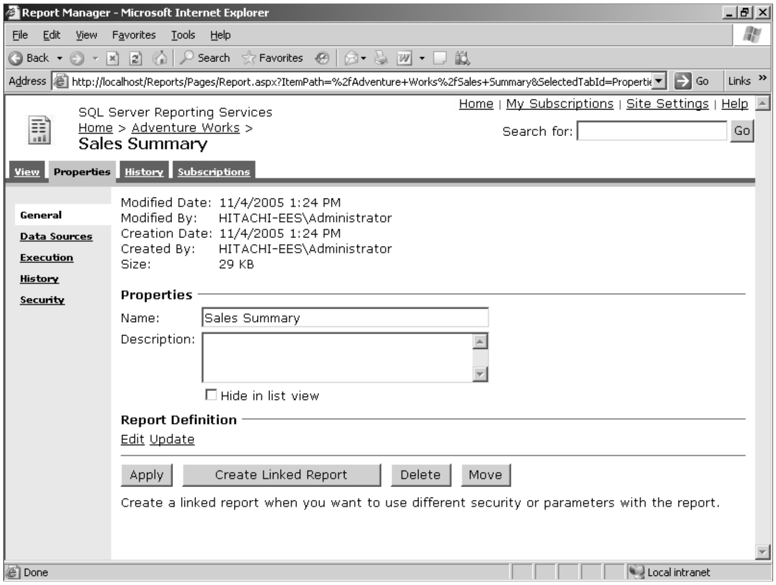
Sales Territory Group	Sales Territory Country	Employee	Actual Sales
North America	Canada	José Saraiva	$1,248,853
		Garrett Vargas	$568,971
	Total		**$1,817,824**
	United States	Shu Ito	$1,068,441
		Linda Mitchell	$1,374,860
		Michael Blythe	$903,230
		Fernando Caro	$1,499,400

The View tab is displayed by default when you open a report. Three other tabs are available for this report: Properties, History, and Subscriptions. In this section, you review the Properties page. Later, in Chapter 7, you'll learn more about the other tabs.

> **Tip** You don't have to wait for the report to be displayed before clicking another tab.

5. Click the Properties tab.

   The Properties page for the Sales Summary report is displayed:


Report Manager - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost/Reports/Pages/Report.aspx?ItemPath=%2fAdventure+Works%2fSales+Summary&SelectedTabId=Properti
Go Links
Home | My Subscriptions | Site Settings | Help
SQL Server Reporting Services
Home > Adventure Works >
Sales Summary
Search for: Go
View Properties History Subscriptions
General
Data Sources
Execution
History
Security
Properties
Report Definition
Edit Update
Apply Create Linked Report Delete Move
Done Local intranet

Modified Date: 11/4/2005 1:24 PM
Modified By: HITACHI-EES\Administrator
Creation Date: 11/4/2005 1:24 PM
Created By: HITACHI-EES\Administrator
Size: 29 KB

Name: Sales Summary
Description:
☐ Hide in list view

Create a linked report when you want to use different security or parameters with the report.

On this page, you can see the author and also the date the report was created. The modification author and date of modification match creation information until the report is subsequently modified. Notice that you can change the name of the report on this page and add a description. The other tasks that you can perform on this page are covered later in Chapter 9.

Notice the links in the left frame of the browser window. The many types of report properties are logically organized into separate pages, where you can apply changes to current settings. Properties determine, for example, how the report appears in Report Manager, how users can interact with the report, and how the Report Server connects to the data sources. You'll review report properties in greater detail in Chapter 9.

Changing Report Properties

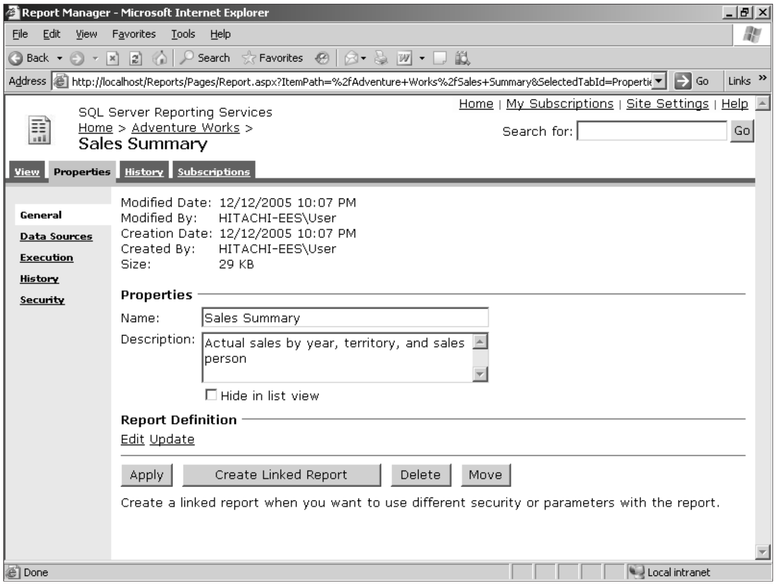
Often, you will want to add a description so a user knows what your report contains before opening it. This property is accessible on the main properties page of the report.

In this procedure, you'll add a description and observe how a description is displayed on the Contents page of a folder.

Add a description

1. In the Description box, type **Actual sales by year, territory, and salesperson**.

   Your screen looks like this:



Report Manager - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Go Links
Home | My Subscriptions | Site Settings | Help
SQL Server Reporting Services
Home > Adventure Works >
Sales Summary
Search for: Go
View Properties History Subscriptions
General
Data Sources
Execution
History
Security
Properties
Report Definition
Edit Update
Apply Create Linked Report Delete Move
Done
Local intranet

Address http://localhost/Reports/Pages/Report.aspx?ItemPath=%2fAdventure+Works%2fSales+Summary&SelectedTabId=Properti

Modified Date: 12/12/2005 10:07 PM
Modified By: HITACHI-EES\User
Creation Date: 12/12/2005 10:07 PM
Created By: HITACHI-EES\User
Size: 29 KB

Name: Sales Summary
Description: Actual sales by year, territory, and sales person
☐ Hide in list view

Create a linked report when you want to use different security or parameters with the report.

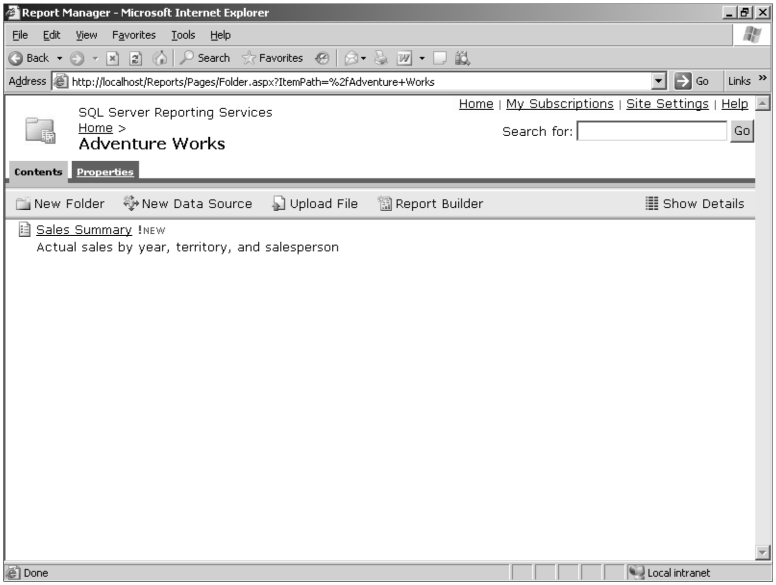
   This report description is displayed on the Contents page and, importantly, is visible only to users who have been granted permission to view the report.

2. Click Apply.

   Clicking Apply doesn't appear to change anything. However, the report description is now visible on the Contents page of the Adventure Works folder.

3. Click the Adventure Works folder link at the top-left corner of the browser window.

Your screen looks like this:


Report Manager - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost/Reports/Pages/Folder.aspx?ItemPath=%2fAdventure+Works
Go Links
Home | My Subscriptions | Site Settings | Help
SQL Server Reporting Services
Home >
Adventure Works
Search for: Go
Contents Properties
New Folder New Data Source Upload File Report Builder Show Details
Done Local intranet

Sales Summary !NEW
Actual sales by year, territory, and salesperson

Notice how the report description is displayed below the report name.

Reviewing Execution Properties

Execution properties are a subset of the report properties maintained for each report. When you understand the implications of the execution property settings, you can choose the most appropriate property setting for your reporting environment. Execution properties allow you to manage reports by balancing system resources and performance with the users' information requirements. For example, you set up caching to achieve a reasonable balance when data used in the report is not changing rapidly at the source. To use caching, you first need to change the data sources properties so you can assign logon credentials that will be used to execute the report for the cache. Separate logon credentials are required by Reporting Services to implement report caching in order to make a single report available to many users.

In this procedure, you'll open the Execution Properties page for your report to review the available options.

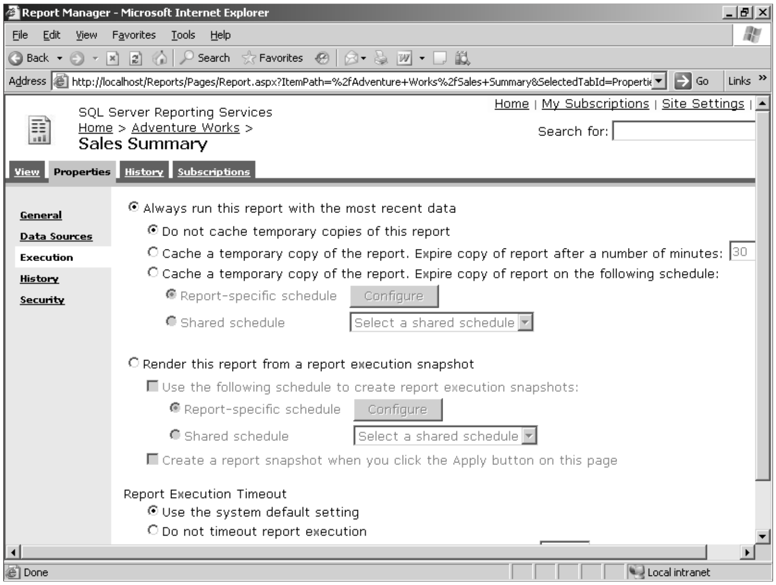
Open the report's Execution Properties page

1. Click the Sales Summary link.

   Assume for a moment that you've just started a new browser session. Just like the previous time you opened this report, a message is displayed to let you know that several activities are occurring: "Report is being generated". Each time a report executes on demand, as you initiated in this step, a query is executed to retrieve data from the rs2005sbsDW database. The data is processed with the report, which is then rendered into the HTML display in your browser window.

2. Click the Properties tab.
3. Click the Execution link in the left frame of the page.

The Execution Properties page is displayed:


Report Manager - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost/Reports/Pages/Report.aspx?ItemPath=%2fAdventure+Works%2fSales+Summary&SelectedTabId=Properti
Go Links
Home | My Subscriptions | Site Settings |
Search for:
View Properties History Subscriptions
General
Data Sources
Execution
History
Security
Configure
Configure
Done
Local intranet

SQL Server Reporting Services
Home > Adventure Works >
Sales Summary

● Always run this report with the most recent data
  ● Do not cache temporary copies of this report
  ○ Cache a temporary copy of the report. Expire copy of report after a number of minutes: 30
  ○ Cache a temporary copy of the report. Expire copy of report on the following schedule:
    ● Report-specific schedule
    ○ Shared schedule: Select a shared schedule

○ Render this report from a report execution snapshot
  ☐ Use the following schedule to create report execution snapshots:
    ● Report-specific schedule
    ○ Shared schedule: Select a shared schedule
  ☐ Create a report snapshot when you click the Apply button on this page

Report Execution Timeout
  ● Use the system default setting
  ○ Do not timeout report execution

Notice that you can render the report either on demand or on a scheduled basis. When rendered on demand, the report typically displays the most recent data and may or may not use caching. By default, a report renders on demand without caching. When rendered on a scheduled basis, the report is stored as an execution snapshot, which captures data for the report at a point in time.

Execution properties control when report processing occurs. When a report executes on demand, which occurs every time another user opens the report, the defined query runs and the query results are merged with the report definition to produce the HTML output format. You can alternatively set the execution properties to process the report at a scheduled time so the report is ready when accessed. However, if you choose to do this, let users know that the data in the report is not current. Another option is to cache a report temporarily to make the same output available to several users for the duration of a specified timeframe. The key difference between a snapshot and a cached report is that the snapshot is stored permanently until physically deleted, while the cache is stored temporarily with a predetermined expiration. The options for specifying execution properties are covered in Chapter 9.

Changing Data Sources Properties

Data sources properties define the connection to be used for query execution. You can change these properties to override the credentials used for authentication when the report executes. Queries that run unattended, such as when a report is cached or scheduled for execution, require stored credentials. These credentials are encrypted when stored in the ReportServer database. (You'll learn more about using secured credentials in Chapter 9.) To set up a cache for your report, you need to change the credentials information in the data sources properties.

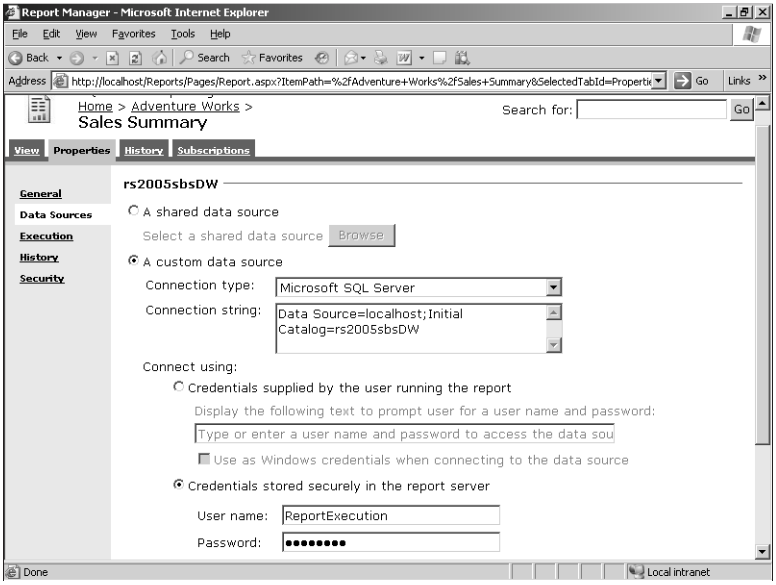
Use secured credentials

1. Click the Data Sources link in the left frame of the page.
2. Click the Credentials Stored Securely In The Report Server option.
3. Type **ReportExecution** as the user name.

   This SQL Server login was added when you installed the sample databases.
4. Type **ReportExecution** as the password.

> **Important** In a production environment in which you are using stored credentials, it's important to test the report by viewing it to ensure that you have entered the user name and password correctly. The credentials will not be validated until the report executes.

Your screen now looks like this:



http://localhost/Reports/Pages/Report.aspx?ItemPath=%2fAdventure+Works%2fSales+Summary&SelectedTabId=Properti

Sales Summary

rs2005sbsDW

○ A shared data source

Select a shared data source

● A custom data source

Connection type: Microsoft SQL Server

Connection string: Data Source=localhost;Initial Catalog=rs2005sbsDW

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 sou

☐ Use as Windows credentials when connecting to the data source

● Credentials stored securely in the report server

User name: ReportExecution

Password: ●●●●●●●●●

5. Click Apply.

Changing Execution Properties

You might want to temporarily cache a report to improve performance. When a user first opens the report, a copy of the report is placed in temporary storage and made available to other users who open the same report. You can also assign a time limit for the cache so that the report can be periodically refreshed with more current data.

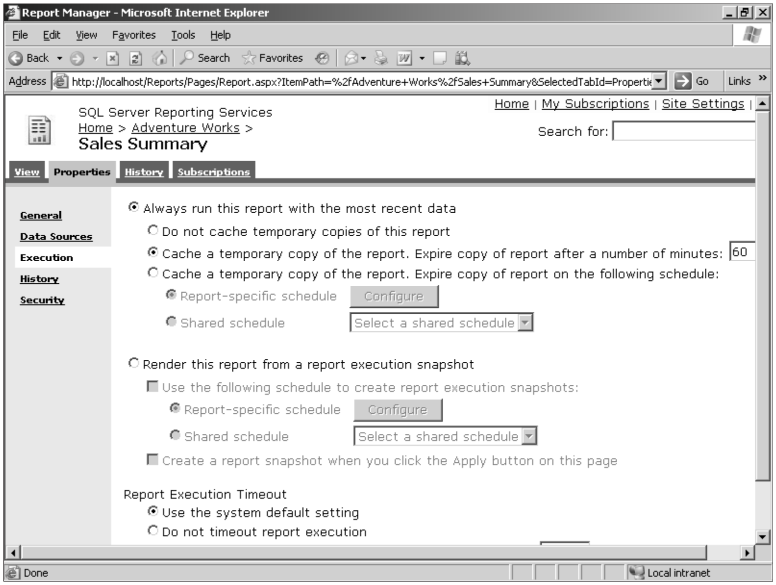
> **Note** The type of caching discussed in this chapter refers to the access of a single report by multiple users. When you open a report, the report is automatically cached for you as part of your browser session. You can then return to this report repeatedly during the same session without having to wait for the query to execute again, regardless of the current setting of the report's execution properties. When you close the browser window, the report is removed from this cache. You'll learn more about session caching in Chapter 9.

In this procedure, you'll change the report execution properties to cache your report, and set the cache to expire after 60 minutes.

Define a report cache

1. Click the Execution link.
2. Click the Cache A Temporary Copy Of The Report. Expire Copy Of Report After A Number Of Minutes option to cache the report and expire after a specified number of minutes. Change the number of minutes to **60**.

   Your screen looks like this:


Report Manager - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost/Reports/Pages/Report.aspx?ItemPath=%2fAdventure+Works%2fSales+Summary&SelectedTabId=Properti
Go Links
Home | My Subscriptions | Site Settings |
Search for:
View Properties History Subscriptions
General
Data Sources
Execution
History
Security
Configure
Select a shared schedule
Done
Local intranet

SQL Server Reporting Services
Home > Adventure Works >
Sales Summary

● Always run this report with the most recent data
- ○ Do not cache temporary copies of this report
- ● Cache a temporary copy of the report. Expire copy of report after a number of minutes: 60
- ○ Cache a temporary copy of the report. Expire copy of report on the following schedule:
  - ● Report-specific schedule
  - ○ Shared schedule

○ Render this report from a report execution snapshot
- ☐ Use the following schedule to create report execution snapshots:
  - ● Report-specific schedule
  - ○ Shared schedule
- ☐ Create a report snapshot when you click the Apply button on this page

Report Execution Timeout
- ● Use the system default setting
- ○ Do not timeout report execution

3. Click Apply at the bottom of the page.

   The next time this report is opened, a temporary copy of it will be placed in the cache to speed up its display for any later requests by other users within the same hour. At the end of 60 minutes, the temporary copy is removed from the cache. A new copy is only placed in the cache when a user requests the report again.

   As mentioned earlier, by using the caching option, you can improve performance for the next user who opens the same report. Any subsequent request for a report results in the display of the cached copy of the report rather than in the execution of the query and processing of the report. That is, any subsequent request displays the cached report until the cache expires. The purpose of expiring the cache on a periodic basis is to force the report to be refreshed with the most current data when the next user accesses the report. The result is a new cached instance of the report until the next scheduled expiration.

> **Tip** The caching feature is useful when you have a query that takes a few minutes or more to execute and many people want to see the same report. It minimizes the demand for resources on the database server, reduces the level of network traffic associated with transporting the data from the database server to the Report Server, and speeds up the display of reports when requested. For more details about report caching, refer to Chapter 9.

Accessing a Report

Each published report has its own URL on the Report Server. Instead of using the Report Manager to navigate through folders to find a report, you can enter the report's URL address into your browser. You can also use this URL in a hyperlink that you add to a custom HTML page. (In fact, you could even include additional characters in the URL to control the behavior of the report, such as formatting the report with a different rendering extension, but you'll learn how to do that in Chapter 18, "Building Custom Reporting Tools.") For now, it's easiest to use the Report Manager to find and view a report online and to export the report to another format.

Displaying a Report

Now that you successfully authored and managed your report, you are ready for the access stage of the reporting life cycle. When you access a report online, you can use a toolbar in the viewer to help you explore your report. After opening the report, you can navigate through its pages or search for specific text so you can jump forward in it.

In this procedure, you'll explore each page of your report.

View report pages

1. Click the View tab.

   The first page of the report, for calendar year 2001, is displayed. This presentation of the report is nearly identical to the version you saw in the Report Designer. You now have the HTML viewer that includes a report toolbar to help you explore and interact with the report. For example, you can use controls in the toolbar to page through the report, to search for a string in the report, or to export the report to another format. The HTML Viewer is covered in more detail in Chapter 12, "Accessing Reports."

2. Click the Next Page button on the View toolbar to view the sales data for each year.

Searching a Report

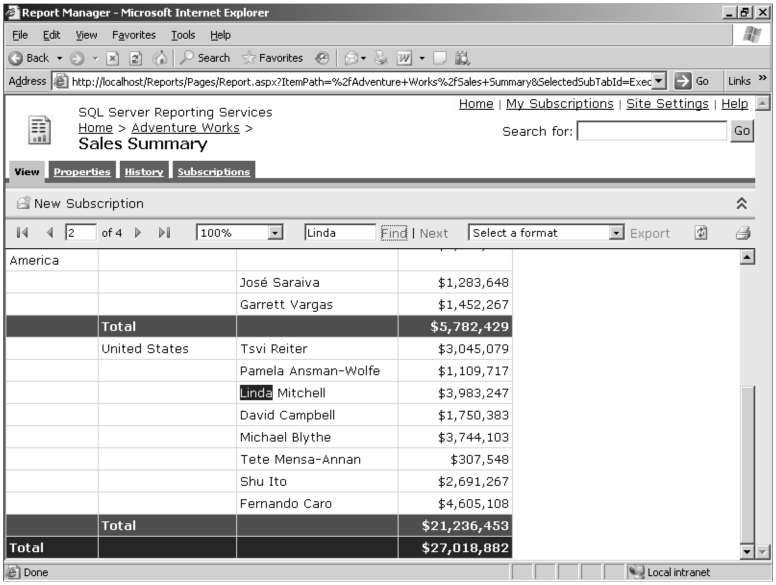
Sometimes the information you're looking for can be difficult to find in a lengthy or multipage report. The HTML Viewer provides a feature to help you find a text string anywhere in the report, from your current position to the end of the report.

In this procedure, you'll use the search feature to locate specific text in the report.

Find text in a report

1. In the Find text box, located in the center of the View toolbar, type **Linda**. Click the Find link.

   Your screen looks like this:


Report Manager - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost/Reports/Pages/Report.aspx?ItemPath=%2fAdventure+Works%2fSales+Summary&SelectedSubTabId=Exec
Go Links
SQL Server Reporting Services
Home > Adventure Works >
Sales Summary
Home | My Subscriptions | Site Settings | Help
Search for:
Go
View
Properties
History
Subscriptions
New Subscription
2 of 4
100%
Linda
Find | Next
Select a format
Export
Done
Local intranet

America			
		José Saraiva	$1,283,648
		Garrett Vargas	$1,452,267
	Total		$5,782,429
	United States	Tsvi Reiter	$3,045,079
		Pamela Ansman-Wolfe	$1,109,717
		Linda Mitchell	$3,983,247
		David Campbell	$1,750,383
		Michael Blythe	$3,744,103
		Tete Mensa-Annan	$307,548
		Shu Ito	$2,691,267
		Fernando Caro	$4,605,108
	Total		$21,236,453
Total			$27,018,882

The Find link is not enabled until you type a string into the associated search box. When you click this link, the report scrolls to the first occurrence of this string in the report.

2. Use the Next link to find the next occurrence of the string.

   The search operation begins in the currently selected page or section and continues across each page of the report until the end of the report is reached.

> **Tip** You don't need to worry about using the correct case, because the search operation is not case-sensitive. However, you are limited to a string length of 256 characters.

Exporting a Report

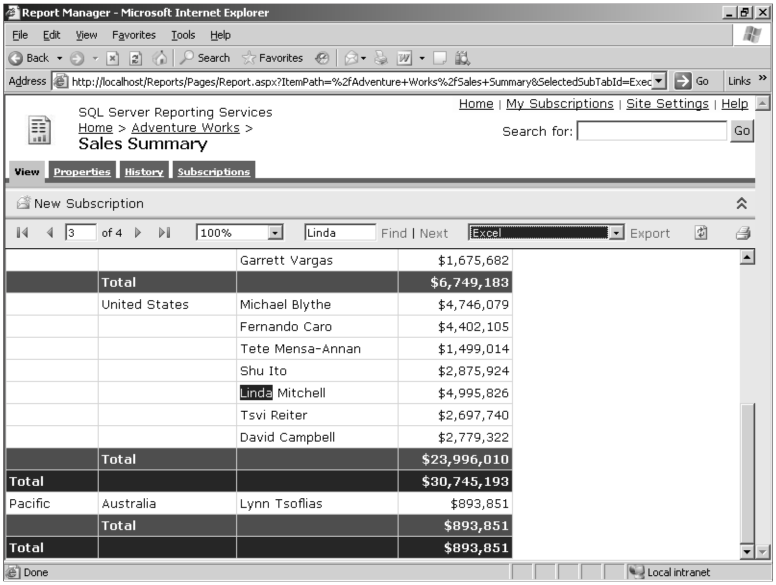
The HTML format is not the only format you can use to view your report. You can also export the report to another format that allows you to create a file that you can open immediately or save to your computer. The View toolbar includes a drop-down list from which you can choose an export format. (Reporting Services includes several export formats, also referred to as rendering formats, which you'll review more closely in Chapter 13, "Rendering Reports.") This feature gives you the flexibility to produce several versions of your report from a single platform.

In this procedure, you'll complete your tour of Reporting Services by exporting your report to an Excel format, opening the generated Excel workbook, and examining each sheet in the workbook, comparing them with the rendering of your report as HTML.

Export to Excel

1. Click Excel in the drop-down list at the far right of the View toolbar.

   Your screen now looks like this:


Report Manager - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Go Links
Home | My Subscriptions | Site Settings | Help
SQL Server Reporting Services
Home > Adventure Works >
Sales Summary
Search for: Go
View Properties History Subscriptions
New Subscription
of 4
Find | Next
Export
Done
Local intranet

Address: http://localhost/Reports/Pages/Report.aspx?ItemPath=%2fAdventure+Works%2fSales+Summary&SelectedSubTabId=Exec

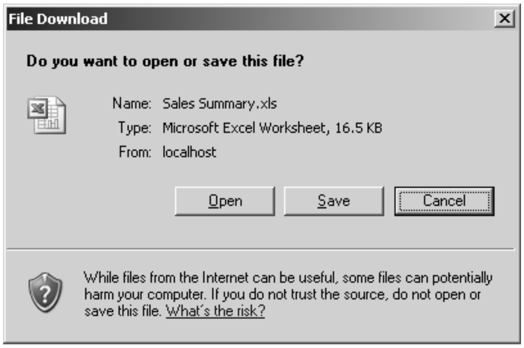
Page: 3 of 4 · 100% · Linda · Excel

		Garrett Vargas	$1,675,682
	Total		$6,749,183
	United States	Michael Blythe	$4,746,079
		Fernando Caro	$4,402,105
		Tete Mensa-Annan	$1,499,014
		Shu Ito	$2,875,924
		Linda Mitchell	$4,995,826
		Tsvi Reiter	$2,697,740
		David Campbell	$2,779,322
	Total		$23,996,010
Total			$30,745,193
Pacific	Australia	Lynn Tsoflias	$893,851
	Total		$893,851
Total			$893,851

As mentioned earlier, the Export feature of the HTML view gives you the ability to view the report in a different format. When you export the report, if a viewer is available for the selected format, a new browser window opens. For example, to export to the Excel format, you must have Microsoft Excel installed on your computer.

2. Click the Export link.

   A new browser window opens and the File Download dialog box is displayed:



Do you want to open or save this file?

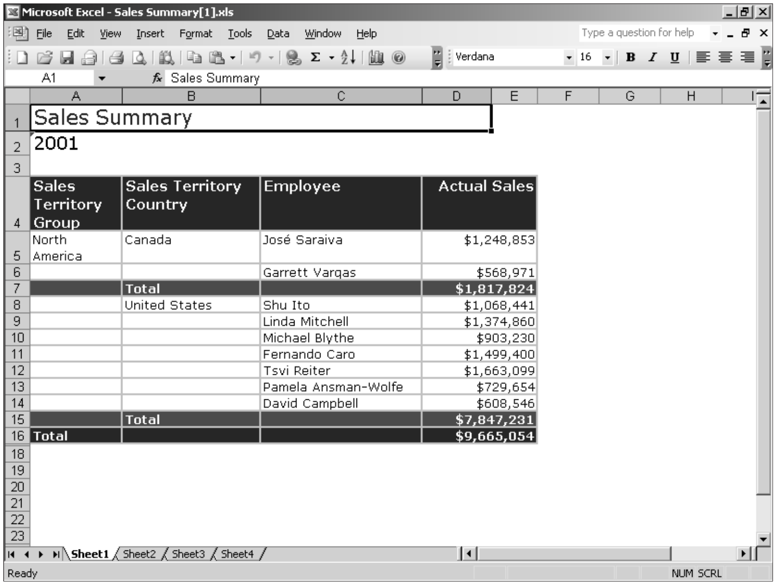
Name: Sales Summary.xls
Type: Microsoft Excel Worksheet, 16.5 KB
From: localhost

While files from the Internet can be useful, some files can potentially harm your computer. If you do not trust the source, do not open or save this file. What's the risk?

   You can open the file to view it now, or you can save the file to view it later.

3. Click Open.

   The Report Server renders the report as an Excel file that downloads to your computer. Microsoft Excel opens, and the report is displayed:



A1: Sales Summary

Sales Summary
2001

Sales Territory Group	Sales Territory Country	Employee	Actual Sales
North America	Canada	José Saraiva	$1,248,853
		Garrett Vargas	$568,971
	Total		$1,817,824
	United States	Shu Ito	$1,068,441
		Linda Mitchell	$1,374,860
		Michael Blythe	$903,230
		Fernando Caro	$1,499,400
		Tsvi Reiter	$1,663,099
		Pamela Ansman-Wolfe	$729,654
		David Campbell	$608,546
	Total		$7,847,231
Total			$9,665,054

Notice that the report style in Reporting Services is closely reproduced in the Excel version of the report. Much of the color style, font style, and layout that you see in the HTML version of the report also appears in the Excel version. Each page has been placed on a separate worksheet in the Excel workbook.

4. Click each worksheet tab to review the sales data for each year.

   With the report in this format, you can take advantage of all of Excel's features to interact with the report data and perform additional analysis that was not possible using the static report in the browser.

Chapter 3 Quick Reference

To	Do this
Start the Report Server Project Wizard	Start a new project in Visual Studio (SQL Server Business Intelligence Development Studio) and select Report Server Project Wizard from the Business Intelligence Projects folder. You must provide a name for the project and solution and designate a folder location for the solution.
Add a data source using the Report Wizard	On the Select The Data Source page, enter a name for the data source; select a connection type; and enter a connection string, or use the Edit button to access the Connection Properties dialog box to generate the connection string automatically. For example: `Data Source=localhost;Initial Catalog=rs2005sbsDW`
Add a query string using the Report Wizard	On the Design The Query page, enter or paste in a query string, or click the Query Builder button to open the Query Builder.
Select a report type using the Report Wizard	On the Select The Report Type page, click either the Tabular or Matrix option.
Arrange the data using the Report Wizard	On the Design The Table page, assign fields to the Page, Group, and Details sections. In the Design The Matrix page, assign fields to the Page, Columns, Rows, and Details sections.
Select a table layout using the Report Wizard	On the Choose The Table Layout page, select Block or Stepped, or optionally include subtotals. If you choose the stepped layout, you can enable drilldown.
Apply a style template using the Report Wizard	On the Choose The Table Style page or the Choose The Matrix Style page, click a style name.
Assign a deployment location and a report name using the Report Wizard	On the Choose The Deployment Location page, enter the URL for the Report Server to host the report. For example: *http://localhost/ReportServer* Optionally, enter a folder name. The folder will be created on deployment if it does not already exist. The final page of the Report Wizard requires a report name.
Preview a report	In Visual Studio, click the Preview tab.

To	Do this
Adjust the size of a column in a table	In Visual Studio, click the table to display the column and row handles, and then drag the column handle to the left to make the column smaller or to the right to make it larger. Alternatively, set the *Width* property for the selected column.
Publish a report solution	On the Build menu of Visual Studio, click Deploy *projectname*.
Open Report Manager	Enter the URL in your browser. For example: *http://localhost/Reports*.
View a report	In Report Manager, navigate the folder hierarchy to the report, and then click the report link.
Manage report properties	With the report open in Report Manager, click the Properties tab. Use the applicable link in the left frame to access the set of properties to be managed. Set a property by clicking an option or selecting a check box, and then clicking the Apply button.
Export a report	With the report open in Report Manager, select the export format from the list box and click the Export link.