



[Technologies](#) > [Reporting Services](#)

Reporting Services Datasheet

Microsoft SQL Server Reporting Services is a comprehensive server-based solution for creating, managing, and delivering traditional, paper-oriented reports as well as interactive, Web-based reports. It complements existing business intelligence and data warehousing features in Microsoft Office, Microsoft Business Solutions, and Microsoft SQL Server.

Regardless of an organization's size or focus, companies leverage corporate assets by empowering employees with real-time information that enables them to make quality business decisions. Microsoft SQL Server Reporting Services improves the productivity of organizations by providing a high-performance, managed reporting environment for the entire enterprise and makes it easier to get the right information to the right people, in virtually any business environment.

An integral part of the Microsoft business intelligence framework, SQL Server Reporting Services combines the data management capabilities of Microsoft SQL Server and Microsoft Windows Server™ with familiar and powerful Microsoft Office System applications to deliver a comprehensive and cost-effective solution for delivering real-time information to support daily operations and decisions.

An Integrated Architecture

With SQL Server Reporting Services, IT developers can seamlessly integrate reporting services into an existing architecture that includes a variety of data sources such as SQL Server, OLE DB, Open Database Connectivity (ODBC), Oracle, and other data providers. Organizations that integrate Reporting Services with existing Microsoft applications, including SQL Server, SQL Server Agent (scheduling), Windows Server 2003, Microsoft Office XP and 2003, SharePoint™ Portal Server 2003, and Visual Studio® .NET 2003 will see accelerated time to value for reporting solutions.

Web Services Architecture

Reporting Services includes a Web-based report server that provides reports and a report management application. By using Web services, Reporting Services can work across a broad array of platforms. This architecture:

- Is both scalable and extensible by design, providing enterprise-class dependability and connectivity.
- Reduces infrastructure and implementation costs.
- Enables quick and easy access to data, improving business productivity and providing superior connectivity between people and information.
- Uses Report Definition Language (RDL), an XML-based industry standard to define reports.

Traditional and Interactive Reports

Reporting Services meets a wide range of reporting needs by supporting both traditional (paper-oriented) and interactive (Web-based) reports. Report developers can customize reports to provide users with the specific information they need in the format that best helps them make more informed decisions.

Integrated Report Authoring

As a comprehensive reporting solution, Reporting Services includes the features and services users need to create and deliver reports. Reporting Services also supports integration with a variety of authoring tools and environments so novice users can create reports without writing a single line of code.

Report Designer, the graphical report development environment included with Reporting Services, uses the Microsoft Visual Studio .NET Integrated Development Environment (IDE) to access all report design features. With Report Designer, IT professionals can define data sources and query information, add data regions and fields to a report, define the report layout, and set up interactive features for reports.

Comprehensive Report Management

Report Manager is a Web application that manages all aspects of the reporting life cycle, including: data sources, report execution and scheduling, report history, security, subscriptions, delivery, and report server administration.

Flexible Report Delivery

Reporting Services enables quick and easy access to the reports users need. With flexible rendering and delivery formats, Reporting Services can also deliver information to a wide variety of devices. Delivery features include:

- **Push or pull delivery.** Reporting Services supports both on-demand (pull) delivery where users request a report dynamically from the report server, as well as scheduled or event-based (push) delivery where subscribers receive reports automatically when a report's data is refreshed or on a specified schedule.
- **Subscriptions.** This feature defines which users want to receive a report, in addition to format and delivery preferences. For example, one user might prefer an HTML version of a report, while another prefers a Microsoft Excel spreadsheet delivered as an e-mail attachment.
- **Personal subscriptions.** Users can create their own, personalized subscriptions, defining preferences for receiving reports as well as which reports they want to receive. Subscriptions are subject to the security model, so users can only subscribe to reports for which they have been granted permissions.
- **Data-driven subscriptions.** In many large organizations, recipients for a report may change frequently as a result of staffing changes or re-organizations. Data-driven subscriptions enable IT professionals to create a dynamically generated list of recipients using an external data source such as an employee database. The reports can also be filtered and personalized using data from the database.
- **Report rendering options.** Because report rendering is separate from the initial processing of the report data, the same report can be delivered in different formats (such as HTML, PDF file, Excel spreadsheet) for different users. These different formats are supported by rendering extensions.

Extensions, APIs, and Interfaces

Reporting Services meets a wide range of reporting needs, including those of independent software vendors (ISVs) and internal IT departments who want to integrate reporting features into tools and applications.

The modular design and extensible application programming interfaces (APIs) enable software developers and enterprises to use SQL Server Reporting Services with legacy systems or non-Microsoft applications. For example, developers can use open interfaces and extensions to:

- Create applications to manage a report server by using a Simple Object Access Protocol (SOAP) interface.
- Create applications or use a Web browser to manipulate report output.
- Create additional rendering, delivery, and data processing extensions by using the Microsoft .NET Framework.

[⤴ Top of page](#)

[Manage Your Profile](#)

© 2007 Microsoft Corporation. All rights reserved. [Terms of Use](#) | [Trademarks](#) | [Privacy Statement](#)

Microsoft