

Microsoft® SQL Server™ 2005

Microsoft SQL Server 2005 Product Overview

Author: Michelle Dumler

Published: April 2005

Summary: This paper provides an overview of the new benefits and functionality available in SQL Server 2005.

Copyright

This is a preliminary document and may be changed substantially prior to final commercial release of the software described herein.

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This White Paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2005 Microsoft Corporation. All rights reserved.

Microsoft, ActiveX, SharePoint, Visual Basic, Visual C#, Visual Studio, Windows, Windows Server, and Windows Server System are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

Table of Contents

Introduction 1

Microsoft SQL Server 2005 Overview 1

SQL Server Data Platform 1

What's New in SQL Server 2005 3

Enterprise Data Management 3

Manageability 4

Availability 5

Scalability 7

Security 8

Developer Productivity 10

Expanded Language Support 10

Improved Development Tools 12

Extensibility 14

Improved Data Access and Web Services 15

Query notification 15

MARS 16

Transparent failover for data mirroring 16

XML Support 16

Application Framework 18

Business Intelligence 22

End-to-End Integrated Business Intelligence Platform 22

Integration Services 23

Analysis Services 24

Reporting Services 24

Integration with the Microsoft Office System 26

Upgrading to SQL Server 2005 26

Microsoft SQL Server Pricing and Licensing 26

Windows Server System Common Engineering Roadmap 28

Conclusion 30

Introduction

Organizations today face numerous data challenges; for example, the need for faster and more data-driven decisions, the need to increase the productivity and flexibility of development staff, and pressure to reduce overall Information Technology budgets while at the same time scaling the infrastructure to meet ever-increasing demands.

The next release of Microsoft® SQL Server™ is designed to help enterprises address these challenges. Microsoft SQL Server 2005 is a next-generation data management and analysis solution that delivers increased security, scalability, and availability to enterprise data and analytical applications, while making them easier to build, deploy, and manage.

Building on the strengths of SQL Server 2000, SQL Server 2005 provides an integrated data management and analysis solution that will help organizations of any size to:

- Build, deploy, and manage enterprise applications that are more secure, scalable, and reliable.
- Maximize Information Technology productivity by reducing the complexity of building, deploying, and managing database applications.
- Share data across multiple platforms, applications, and devices to make it easier to connect internal and external systems.
- Control costs without sacrificing performance, availability, scalability, or security.

Read on to learn more about SQL Server 2005 advancements in three key areas: enterprise data management, developer productivity, and business intelligence as well pricing and licensing, upgrading to SQL Server 2005, and the Microsoft Windows Server System™.

Microsoft SQL Server 2005 Overview

SQL Server Data Platform

SQL Server is a comprehensive, integrated end-to-end data solution that empowers users across your organization by providing them with a secure, reliable, and productive platform for enterprise data and business intelligence (BI) applications. SQL Server 2005 delivers powerful, familiar tools to Information Technology professionals as well as to information workers, reducing the complexity of creating, deploying, managing, and using enterprise data and analytical applications on platforms ranging from mobile devices to enterprise data systems. Through a comprehensive feature set, interoperability with existing systems, and automation of routine tasks, SQL Server 2005 provides a complete data solution for enterprises of all sizes. Figure 1 shows the layout of the SQL Server 2005 data platform.

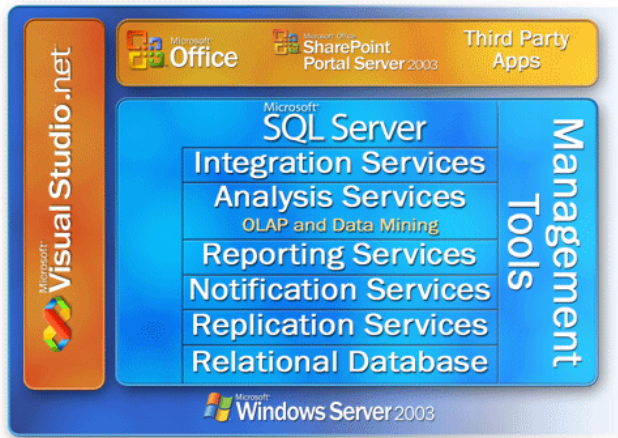


Figure 1: The SQL Server 2005 Data Platform

The SQL Server data platform includes the following tools:

- **Relational database:** Secure, reliable, scalable, highly available relational database engine with improved performance and support for structured and unstructured (XML) data.
- **Replication Services:** Data replication for distributed or mobile data processing applications, high systems availability, scalable concurrency with secondary data stores for enterprise reporting solutions, and integration with heterogeneous systems, including existing Oracle databases.
- **Notification Services:** Advanced notification capabilities for the development and deployment of scalable applications that can deliver personalized, timely information updates to a variety of connected and mobile devices.
- **Integration Services:** Extract, transform, and load capabilities for data warehousing and enterprise-wide data integration.
- **Analysis Services:** Online analytical processing (OLAP) capabilities for the rapid, sophisticated analysis of large and complex datasets using multidimensional storage.
- **Reporting Services:** A comprehensive solution for creating, managing, and delivering both traditional, paper-oriented reports and interactive, Web-based reports.
- **Management tools:** SQL Server includes integrated management tools for advanced database management and tuning as well as tight integration with tools such as Microsoft Operations Manager (MOM) and Microsoft Systems Management Server (SMS). Standard data access protocols drastically reduce the time it takes to integrate data in SQL Server with existing systems. In addition, native Web service support is built into SQL Server to ensure interoperability with other applications and platforms.

-
- **Development tools:** SQL Server offers integrated development tools for the database engine, data extraction, transformation, and loading (ETL), data mining, OLAP, and reporting that are tightly integrated with Microsoft Visual Studio® to provide end-to-end application development capabilities. Every major subsystem in SQL Server ships with its own object model and set of APIs to extend the data system in any direction that is unique to your business.

The SQL Server 2005 data platform provides organizations of all sizes with the following benefits:

- **Leverage data assets:** In addition to delivering a secure, reliable database for line-of-business and analytical applications, SQL Server 2005 enables customers to get more value from their data by including embedded functionality such as reporting, analysis, and data mining.
- **Increase productivity:** Through comprehensive business intelligence capabilities and integration with familiar tools such as the Microsoft Office System, SQL Server 2005 provides information workers across your organization with critical, timely business information that is tailored to their specific needs. The goal is to extend BI to all users within an organization and ultimately to allow users at all levels of the organization to make better business decisions based on one of their most valuable assets—their data.
- **Reduce Information Technology complexity:** SQL Server 2005 simplifies the development, deployment, and management of line-of-business and analytical applications by providing a flexible development environment for developers and integrated, automated management tools for database administrators.
- **Lower total cost of ownership (TCO):** The integrated approach and focus on ease-of-use and deployment provides the industry's lowest upfront, implementation, and maintenance costs for rapid return on your database investment.

What's New in SQL Server 2005

Enterprise Data Management

In today's connected world, data and the systems that manage that data must always be secure yet available to your users. With SQL Server 2005, users and Information Technology professionals across your organization will benefit from reduced application downtime, increased scalability and performance, and tight yet flexible security controls. SQL Server 2005 also includes many new and improved capabilities to help make your Information Technology staff more productive. SQL Server 2005 includes key enhancements to enterprise data management in the following areas:

- Manageability
- Availability
- Scalability
- Security

Manageability

SQL Server 2005 makes it simpler and easier to deploy, manage, and optimize enterprise data and analytical applications. As an enterprise data management platform, it provides a single management console that enables data administrators anywhere in your organization to monitor, manage, and tune all of the databases and associated services across your enterprise. It provides an extensible management infrastructure that can be easily programmed using SQL Management Objects (SMO), enabling users to customize and extend their management environment and Independent Software Vendors (ISVs) to build additional tools and functionality to further extend the capabilities that come out of the box.

SQL Server Management Studio

SQL Server 2005 simplifies management by providing one integrated management console to monitor and manage the SQL Server relational database, as well as Integration Services, Analysis Services, Reporting Services, Notification Services, and SQL Mobile across large numbers of distributed servers and databases. Database administrators can perform several tasks at the same time including: authoring and executing a query, viewing server objects, managing an object, monitoring system activity, and viewing online help. SQL Server Management Studio hosts a development environment for authoring, editing and managing scripts and stored procedures using Transact-SQL, Multidimensional Expressions (MDX), XMLA, and SQL Server Mobile Edition. Management Studio is readily integrated with source control. Management Studio also hosts tools for scheduling SQL Server Agent jobs and managing maintenance plans to automate daily maintenance and operation tasks. The integration of management and authoring in a single tool coupled with the ability to manage all types of servers provides enhanced productivity for database administrators.

“We have thousands of stored procedures, and with SQL Server 2000 I used to have to use a separate tool to check out code, and then open the query analyzer to edit the code. With SQL Server 2005 all of this is integrated with Management Studio. I’m able to accomplish routine tasks 20 percent faster using Management Studio.”

***--Joyce Behrendt, Senior Development Manager, Information Technology
Corporate Strategy Planning and Analysis, Microsoft***

Proactive Performance Monitoring and Performance Tuning

SQL Server 2005 exposes more than 70 new measures of internal database performance and resource usage from memory, locking, and scheduling to transactions and network and disk I/O. These Dynamic Management Views (DMVs) provide greater transparency and visibility into the database and a powerful infrastructure for proactive monitoring of database health and performance.

SQL Management Objects

SQL Management Objects (SMO) is a new set of programming objects that exposes all of the management functionality of the SQL Server database. In fact, Management Studio was built with SQL Management Objects. SMO is implemented as a Microsoft .NET Framework assembly. You can use SMO to automate common SQL Server administrative tasks, such as programmatically retrieving configuration settings, creating new databases, applying Transact-SQL scripts, creating SQL Server Agent jobs, and scheduling backups. The SMO object model is a more secure, reliable, and scalable replacement for Distributed Management Objects (DMO), which was included with earlier versions of SQL Server.

Availability

Investments in high availability technologies, additional backup and restore capabilities, and replication enhancements will enable enterprises to build and deploy highly available applications. Innovative high availability features such as; database mirroring, failover clustering, database snapshots, and enhanced online operations will minimize downtime and help to ensure that critical enterprise systems remain accessible. We will review these enhancements in greater detail in this section.

Database Mirroring

Database mirroring allows continuous streaming of the transaction log from a source server to a single destination server. In the event of a failure of the primary system, applications can immediately reconnect to the database on the secondary server. The secondary instance detects failure of the primary server within seconds and accepts database connections immediately. Database mirroring works on standard server hardware and requires no special storage or controllers. Figure 2 shows the basic configuration of database mirroring.

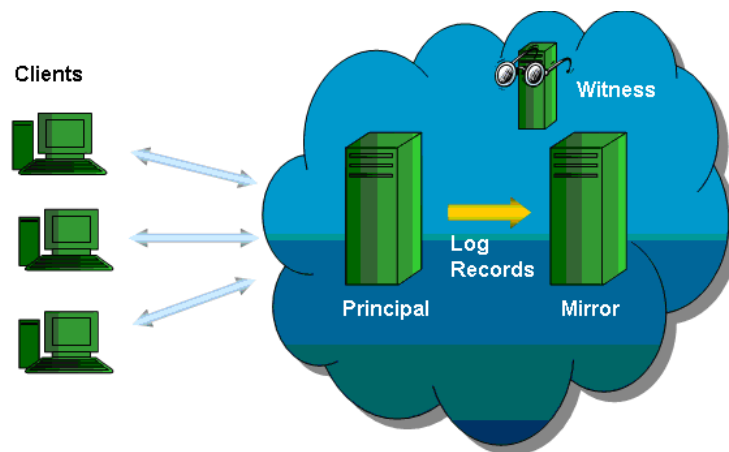


Figure 2: Basic Configuration of Database Mirroring

Failover Clustering

Failover clustering is a high availability solution that exploits Microsoft Windows® Clustering Services to create fault-tolerant virtual servers that provide fast failover in the event of a database server failure. In SQL Server 2005, support for failover clustering has been extended to SQL Server Analysis Services, Notification Services, and SQL Server replication. The maximum number of cluster nodes has been increased to eight. SQL Server failover clustering is now a complete fault-tolerant server solution.

Availability Feature	Database Mirroring	Failover Clustering
Automatic Failover	Yes	Yes
Transparent Client Redirection	Yes, auto-redirect	Yes, reconnect to same IP
Impact on Overall Throughput	No impact to minimal	No impact
Zero Work Loss	Yes	Yes
Requires Certified Hardware	No	Yes
Provides Redundant Data	Yes	No

Database Snapshots

SQL Server 2005 introduces the ability for database administrators to create instant, read-only views of a database. The database snapshot provides a stable view without the time or storage overhead of creating a complete copy of the database. As the primary database diverges from the snapshot, the snapshot adds its own copy of pages as they are modified. Thus the snapshot may be used to quickly recover from an accidental change to a database by simply reapplying the original pages from the snapshot to the primary database.

Fast Recovery

SQL Server 2005 improves the availability of SQL Server databases with a new faster recovery option. Users can reconnect to a recovering database after the transaction log has been rolled forward. Earlier versions of SQL Server required users to wait until incomplete transactions had rolled back, even if the users did not need to access the affected parts of the database.

Dedicated Administrator Connection

SQL Server 2005 introduces a dedicated administrator connection to access a running server even if the server is not responding or is otherwise unavailable. This allows you to execute diagnostic functions or Transact-SQL statements in order to troubleshoot problems on a server. The connection is activated by members of the **sysadmin** fixed server role and is only available through the SQLCMD command prompt utility either locally, or from a remote machine.

Online Operations (index operations and restore)

The ability to create, rebuild, or drop an index online is an enhanced feature of SQL Server 2005 that augments the indexing capabilities of earlier versions of SQL Server. The online index option allows concurrent modifications (updates, deletes, and inserts) to the underlying table or clustered index data and any associated indexes during index data definition language (DDL) execution. With support for online index operations, you can add indexes without interfering with access to tables or other existing indexes. Additionally, the server workload allows index operations to take advantage of parallel processing

SQL Server 2005 also introduces the ability to perform a restore operation while an instance of SQL Server is running. Online restoration capabilities improve the availability of SQL Server because only the data that is being restored is unavailable. The rest of the database remains online and available. Earlier versions of SQL Server require that you bring a database offline before you restore the database.

Replication

Replication is designed to increase data availability by distributing the data across multiple database servers. Availability is increased by allowing applications to scale out the SQL Server read workload across databases. SQL Server 2005 offers enhanced replication using a new peer-to-peer model that provides a new topology in which databases can be synchronized transactionally with any identical peer database.

Scalability

Scalability advancements such as table partitioning, snapshot isolation, and 64-bit support will enable you to build and deploy your most demanding applications using SQL Server 2005. The partitioning of large tables and indexes significantly enhances query performance against very large databases.

Table and Index Partitioning

Table and index partitioning eases the management of large databases by facilitating the management of the database in smaller, more manageable chunks. While the concept of partitioning data across tables, databases, and servers is not new to the world of databases, SQL Server 2005 provides a new capability for the partitioning of tables across filegroups in a database. Horizontal partitioning allows for the division of a table into smaller groupings based on a partitioning scheme. Table partitioning is designed for very large databases, from hundreds of gigabytes to terabytes and beyond.

Snapshot Isolation

Once data is copied, transformed, and archived to an analysis-oriented database it must be maintained and/or rebuilt periodically. Users certainly benefit from looking at a transactionally consistent version of the database; however, the version of the data that they are viewing is no longer current. It can take many hours to build and index the and that might not be what the user really needs. This is where snapshot isolation comes in. The snapshot isolation level allows users to access the last row that was committed, by using a transactionally consistent view of the database. This new isolation level provides the following benefits:

- Increased data availability for read-only applications.
- Nonblocking read operations allowed in an OLTP environment.
- Automatic mandatory conflict detection for write transactions.
- Simplified migration of applications from Oracle to SQL Server.

Replication Monitor

Replication Monitor is a tool that sets a new standard for ease of use in managing complex data replication operations with its intuitive user interface and wealth of data metrics.

Support for 64-Bit System Itanium2 and x64

Optimized for the Intel Itanium processor, SQL Server (64-bit) takes advantage of advanced memory addressing capabilities for essential resources such as buffer pools, caches, and sort heaps, reducing the need to perform multiple I/O operations to bring data in and out of memory from disk. Greater processing capacity without the penalties of I/O latency opens the door to new levels of application scalability.

Windows Server™ 2003 x64 provides high performance™ for both 32-bit and 64-bit applications on the same system. The underlying architecture is based on 64-bit extensions to the industry-standard x86 instruction set, allowing today's 32-bit applications to run natively on x64 processors. At the same time, new 64-bit applications are executed in 64-bit mode, which processes more data per clock cycle, allows greater access to memory, and speeds numeric calculations. The end result is a platform that leverages the existing wealth of 32-bit applications while also providing a smooth migration path to 64-bit computing.

Security

SQL Server 2005 makes significant enhancements to the security model of the database platform, with the intention of providing more precise and flexible control to enable tighter security of the data. A considerable investment has been made in a number of features to provide a high level of security for your enterprise data including:

- Enforcing policies for SQL Server login passwords in the authentication space.
- Providing for more granularity in terms of specifying permissions at various scopes in the authorization space.
- Allowing for the separation of owners and schemas in the security management space.

Authorization

A new security model in SQL Server 2005 allows administrators to manage permissions at a granular level and at a designated scope, making management of permissions easier as well as ensuring that the principle of least privileges is upheld. SQL Server 2005 allows you to specify a context under which statements in a module execute. This feature also acts as an excellent mechanism for granular permission management.

Authentication

SQL Server 2005 clustering supports Kerberos authentication against a SQL Server 2005 virtual server. Administrators can specify Microsoft Windows-style policies on standard logins so that a consistent policy is applied across all accounts in the domain.

Native Encryption

SQL Server 2005 supports encryption capabilities within the database itself, fully integrated with a key management infrastructure. By default, client/server communications are encrypted. To centralize security assurance, server policy can be defined to reject unencrypted communications.

SQL and Trustworthy Computing

The Trustworthy Computing initiative outlines a framework that defines the steps necessary to support secure computing as well as measures that help you deploy and maintain a secure environment. These steps help to protect the confidentiality, integrity, and availability of data and systems at every phase of the software life cycle—from design, to delivery, to maintenance. To uphold the four tenets of the Trustworthy Computing initiative, Microsoft and the SQL Server team have taken the following steps:

- **Secure by design.** The SQL Server development team conducted multiple security audits and spent more than two months studying SQL Server components and the interaction between them. For each potential security threat, the team did a threat analysis to evaluate the issue and completed additional design and testing work to neutralize potential security issues. As a result of these design efforts, SQL Server 2005 includes many new server security features.
- **Secure by default.** Upon installation, SQL Server 2005 chooses the right set of configuration values for all setup options, ensuring that when a new system is installed, it will be in a secure state by default.
- **Secure in deployment.** Microsoft has created content to help organizations deploy SQL Server using the proper security credentials and to fully understand the steps and permissions required. SQL Server deployment tools provide the information necessary to understand the decisions you need to make during deployment. Security updates are easy to find and install—and if you choose the option, the updates install automatically. Tools are also available to help you assess and manage security risks across organizations.

Developer Productivity

SQL Server 2005 includes many new technologies that bring significant increases in developer productivity. From .NET Framework support to tight integration with Visual Studio®, these features provide developers with the ability to more easily create secure, robust database applications at a lower cost. SQL Server 2005 enables developers to leverage existing skills across a variety of development languages while providing an end-to-end development environment for the database. Native XML capabilities will also allow developers to build new classes of connected applications across any platform or device.

Enhancements for developer productivity include:

- Expanded language support
- Improved development tools
- Extensibility
- Improved data access
- XML and Web services
- Application Framework

Expanded Language Support

Because the common language runtime (CLR) is hosted in the database engine, developers can choose from a variety of familiar languages to develop database applications, including Transact-SQL, Microsoft Visual Basic® .NET, and Microsoft Visual C#® .NET. Additionally, CLR hosting will provide developers with increased flexibility through the use of user-defined types and functions. The CLR will also provide opportunities to use third-party code for rapid database application development.

CLR/.NET Framework integration

With the release of Microsoft SQL Server 2005, database programmers can now take full advantage of the Microsoft .NET Framework class library and modern programming languages to implement functionality within the server. Using common language runtime (CLR) integration, you can code your stored procedures, functions, and triggers in the .NET Framework language of your choice. Microsoft Visual Basic .NET and the C# programming language both offer object-oriented constructs, structured exception handling, arrays, namespaces, and classes. In addition, the .NET Framework provides thousands of classes and methods that have extensive built-in capabilities that you can easily use on the server-side. Many tasks that were awkward or difficult to perform in Transact-SQL can be better accomplished by using managed code; additionally, two new types of database objects—aggregates and user-defined types—are available. You can now better use the knowledge and skills that you have already acquired to write in-process code. In short, SQL Server 2005 enables you to extend the database server to more easily perform appropriate computation and operations on the back end.

This integration between SQL Server and the CLR provides several major benefits:

- **Enhanced programming model:** Programming languages that are compatible with the .NET Framework are in many respects richer than Transact-SQL, offering constructs and capabilities that were previously not available to SQL developers.
- **Enhanced safety and security:** Managed code runs in a CLR environment, hosted by the database engine. This allows .NET Framework database objects to be safer and more secure than the extended stored procedures available in earlier versions of SQL Server.
- **User-defined types and aggregates:** Two new database objects that expand the storage and querying capabilities of SQL Server are enabled by hosting the CLR.
- **Common development environment:** Database development is integrated into the Microsoft Visual Studio 2005 development environment. You can use the same tools for developing and debugging database objects and scripts that you use to write middle-tier or client-tier .NET Framework components and services.
- **Performance and scalability:** Because managed code compiles to native code prior to execution, you can achieve significant performance increases in some scenarios.

By using languages such as Visual Basic .NET and C#, you can capitalize on CLR integration to write code that has more complex logic and is more suited for computation tasks. In addition, Visual Basic .NET and C# offer object-oriented capabilities such as encapsulation, inheritance, and polymorphism. You can easily organize related code into classes and namespaces, which means that you can more easily organize and maintain your code investments when you are working with large amounts of code. The ability to logically and physically organize code into assemblies and namespaces is a huge benefit, that allows you to better find and relate different pieces of code in a large database implementation.

Managed code is more efficient than Transact-SQL at processing numbers and managing complicated execution logic, and provides extensive support for string handling, regular expressions, and so on. Also, with the functionality that is available in the .NET Framework class library, you have full access to thousands of pre-built classes and routines that you can access easily from any stored procedure, trigger, or user-defined function. Everything from improved string-handling functions, math functions, date operations, access to system resources, advanced encryption algorithms, file access, image processing, and XML data manipulation is easily accessible from managed stored procedures, functions, triggers, and aggregates.

One of the major benefits of managed code is type safety. Before managed code is executed, the CLR performs several checks through a process known as *verification* to ensure that the code is safe to run. For example, the code is checked to ensure that memory is not read that has not been written to.

Transact-SQL enhancements

Transact-SQL has long been the basis for all programmability of SQL Server. SQL Server 2005 provides many new language capabilities for developing scalable database applications. These enhancements include error handling, new recursive query capabilities, and support for new SQL Server Database Engine capabilities. Transact-SQL enhancements in SQL Server 2005 increase your expressive powers in query writing, allowing you to improve the performance of your code and extend your error management capabilities. The continuous effort that is being put into enhancing Transact-SQL shows a firm belief in its significant role in SQL Server.

Improved Development Tools

Developers will be able to use one development tool for Transact-SQL, XML, Multidimensional Expressions (MDX), and XML for Analysis (XML/A). Integration with the Visual Studio development environment will provide more efficient development and debugging of line-of-business and business intelligence (BI) applications.

Business Intelligence Development Studio

The Business Intelligence Development Studio is a common development environment for building BI solutions based on Visual Studio including a database engine, analysis services, and reporting services. Use the Business Intelligence Development Studio graphical user interface to design SQL Server Integration Services (SSIS) packages for data management applications. SSIS packages are designed, developed, and debugged in the Business Intelligence Development Studio by dragging tasks from the toolbox, setting their properties, and connecting tasks with precedence constraints. Figure 3 shows the interface in Visual Studio for the Business Intelligence Development Studio.

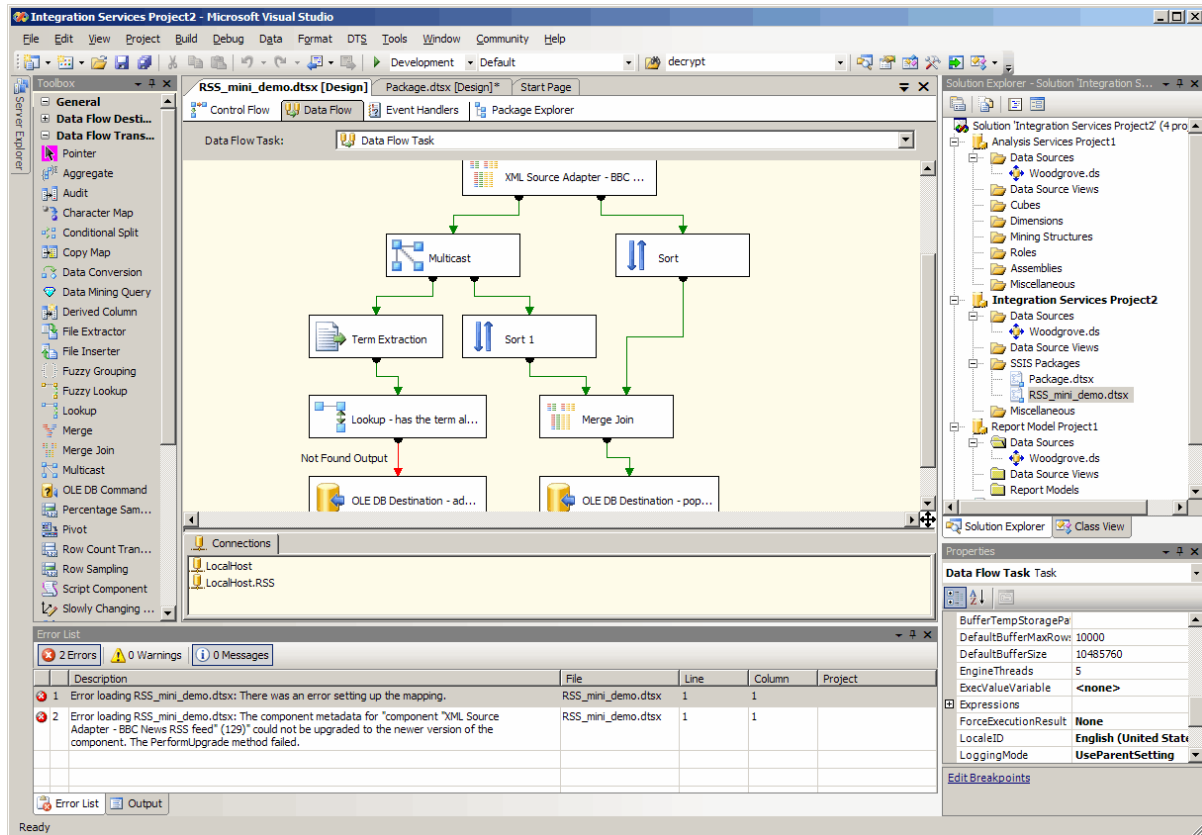


Figure 3: Business Intelligence Development Studio interface in Visual Studio

Visual Studio Integration

SQL Server 2005 and Visual Studio 2005 together provide deeper levels of integration between the database and the application development environment than ever before. Developers now have the ability to create CLR stored procedures, functions, user-defined types, and user-defined aggregates directly from within the Visual Studio development environment. They can deploy these new database objects directly from Visual Studio without having to switch tools. Visual Studio 2005 supports all of the new SQL Server data types, such as native XML directly. You can also add your CLR database objects to the same source control system that you use for all your Visual Studio projects, thus providing an even greater level of integration and security to your development processes.

Cross-Tier and Cross-Language Debugging

SQL Server 2005 and Visual Studio 2005 together provide even deeper integration in the area of application debugging. The combination allows you to seamlessly debug both CLR and Transact-SQL code using the same Visual Studio debugging interface and it allows you to debug from CLR to Transact-SQL and back again, regardless of the location of the code, whether it is on the developer machine or stored in the SQL Server database.

Extensibility

User-Defined Types and Aggregates

User-defined types in SQL Server 2005 are not an object relational extensibility mechanism. They are a way to extend the scalar type system of the database. The scalar type system includes the columnar types that ship with SQL Server (types like **int**, **nvarchar**, **uniqueidentifier**, etc.). With user-defined types, you can define your own type that can be used for column definitions, for example. Create a user-defined type if your type really is an atomic value that is appropriate to be modeled as a column.

Use user-defined types if you need to define your own scalar type. Example scenarios for such types include custom date/time data types in various calendars, and currency data types. With user-defined types, you can create a single object that exposes all the behaviors that are available on the type, and encapsulate, or hide, the underlying data that is stored by the type. Everyone that needs to access the data has to use the user defined type programmatic interface. If you can leverage existing functionality in the .NET Framework (such as the internationalization or calendar functionality), that is another really good reason to consider implementing your type as a user-defined type.

There are a number of scenarios where you may need to perform aggregations over data. This includes performing statistical calculations, such as avg, stddev, etc. If the desired aggregation function is not directly supported as a built-in aggregate function, there are three ways to perform a custom aggregation in SQL Server 2005:

- Write the aggregation as a user-defined aggregate.
- Write the aggregate using a CLR stored procedure.
- Use a server-side cursor.

SQL Management Objects(SMO)

SQL Management Objects (SMO) is the management object model for SQL Server 2005. SMO represents significant design and architectural improvements for the SQL Server management object model. It is a simple to use but rich object model that is based on .NET Framework managed code. SMO is the primary tool for developing database management applications using .NET Framework. SMO is used by every dialog box in SQL Server Management Studio, and every administrative action that you can perform in SQL Server Management Studio you can also accomplish by using SMO.

The new SMO object model and the Microsoft Windows Management Instrumentation (WMI) APIs replace SQL-DMO. Where possible, SMO incorporates similar objects as SQL-DMO for ease of use. You can still use SQL Server 2005 with SQL-DMO, but SQL-DMO will not be updated to manage features that are specific to SQL Server 2005.

Analysis Management Objects

Analysis Management Objects (AMO) allows client applications to access the range of administrative commands and capabilities available to Analysis Services using an object library that can provide object-level validation capabilities, instead of having to manually generate DDL scripts for Analysis Services commands and the often-lengthy contents of the Analysis Services Scripting Language (ASSL) **ObjectDefinition** element. Applications using AMO can either connect and work directly with objects on an Analysis Services instance, or create such objects without an existing connection and persist the metadata for later deployment. AMO also “wraps” ASSL commands and elements.

Improved Data Access and Web Services

In SQL Server 2005, you can develop XML Web services in the database tier, making SQL Server an HTTP listener. This provides a new type of data access capability for applications that are centralized around Web services. In SQL Server 2005 you can use HTTP to access SQL Server directly, without using a middle-tier listener such as Microsoft Internet Information Services (IIS). SQL Server exposes a Web service interface, to allow the execution of SQL statements and invocation of functions and procedures. Query results are returned in XML format and can take advantage of the Web services infrastructure of Visual Studio.

ADO.NET 2.0/ADOMD.NET

There is much that's new in the next version of ADO.NET. From new support for query change notifications, to Multiple Active Result Sets (MARS), ADO.NET evolves dataset access and manipulation to achieve greater scalability and flexibility.

Query notification

SQL Server 2005 introduces notification support for SQL Server queries. You can use this support to send a command to SQL Server and to request that a notification be generated if executing the same command again produces different results from those obtained initially. You accomplish this by using a dependency object that detects when the underlying data is changed. Commands that are sent to the server through any of the client APIs such as ADO.NET, OLE DB, Open Database Connectivity (ODBC), Microsoft ActiveX® Data Objects (ADO), or SOAP may include a tag that requires a notification. For each statement that is executed as part of the request, the server creates a *notification subscription* that fires once for each statement that is included in the request. Notifications are delivered through a SQL Service Broker queue that applications can poll, and use activation services or blocking statements that return whenever the notifications are available. Query notifications are useful for enabling the caching of results in applications such as database-driven Web sites. Figure 4 shows the query notification process.

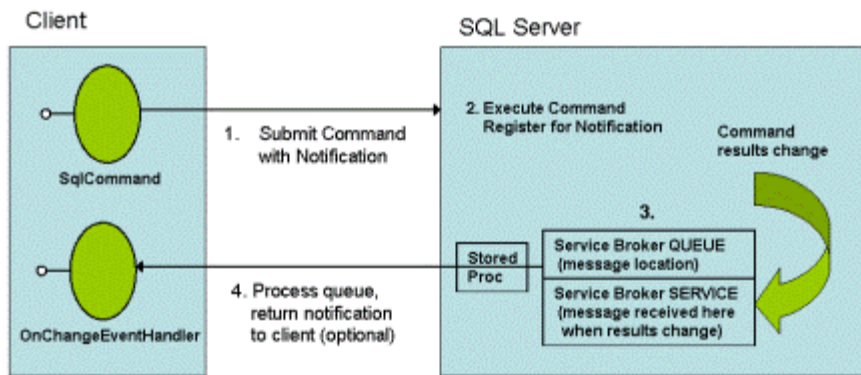


Figure 4: Query notification

MARS

MARS provides the ability to have more than one pending request per connection, in particular to have more than one default result set open per connection. Default result sets are forward-only read-only result sets. For default result sets, the client drivers transparently retrieve the data in large chunks (Tabular Data Stream buffer sized chunks) so that application requests are satisfied without a roundtrip to the server (as in the case of server cursors). The application can use a simple row-at-a-time programming model without compromising performance. The multiple active result sets feature removes the current restriction in which an open default result set blocks the driver from sending requests to the server until the entire result set is consumed.

Transparent failover for data mirroring

SQL Server 2005 supports a "hot spare" capability through database mirroring. If a SQL Server instance fails, the work can be shifted over to the backup server automatically. This requires an instance to witness the failover known as (not surprisingly) the *witness instance*. Hot spare scenarios require that existing client connections must "know" to fail over (establish a connection with the new server instance), as well. Client connections that produce an error on the next attempted access and must be manually "failed over" by client programming are suboptimal. SqlClient in ADO.NET 2.0 supports client failover without special programming of the application program.

XML Support

Advancements such as native XML data type and XQuery help organizations to seamlessly connect internal and external systems. SQL Server 2005 will support both relational and XML data natively, so enterprises can store, manage, and analyze data in the format that best suits their needs. Support for existing and emerging open standards such as Hypertext Transfer Protocol (HTTP), XML, Simple Object Access Protocol (SOAP), XQuery, and XML Schema definition language (XSD) will also facilitate communication across extended enterprise systems.

XML Data Type

XML can model complex data; it is not limited to the scalar types that are supported by SQL Server. As such, a string-based, built-in data type such as **char** or **varchar** does not suffice to make full and effective use of the power and the numerous advantages of XML. For example, if XML is stored as a string, you can insert or select an entire document, or even retrieve contiguous bytes from it, but you cannot query into the contents of the document itself. By providing the XML data type, SQL Server 2005 allows you to query portions of an XML document, validate that the document conforms to an XML schema, and even modify the contents of the XML document in place. It also integrates traditional, relational data with data in unstructured or semi-structured XML documents in ways that are not possible with SQL Server 2000. In SQL Server 2005, XML data is stored as binary large objects (BLOBs) in an internal representation that allows efficient reparsing and some compression.

A collection of XML schemas can be associated with a column of type XML. This provides validation for constraints, inserts, and updates, and typing of values inside stored XML data, as well as optimizations for storage and query processing. SQL Server 2005 also provides several DDL statements for managing schemas on the server.

XQuery

The XML Query Language, or XQuery, is an intelligent and robust language that is optimized for querying all types of XML data. With XQuery you can run queries against variables and columns of the XML data type using the latter's associated methods. As with many of the XML standards, the World Wide Web Consortium (W3C) oversees the development of XQuery. XQuery evolved from a query language called Quilt, which was itself based on a variety of other query languages such as the XML Path Language (XPath) version 1.0, XQL, and SQL. It also contains XPath 2.0 as a subset. Therefore, if you have experience using XPath 1.0, you can capitalize on your skills and do not have to learn an entirely new query language. There are, however, significant enhancements that go beyond XPath 1.0, such as typing, special functions, and support for better iteration, sorting of results, and construction.

SQL Server 2005 ships with deep XQuery capabilities that allow for XML object manipulation in the data tier. It supports a statically typed subset of the XQuery 1.0 Working Draft of November 15, 2003.

Web Services Support

In SQL Server 2005, you can develop XML Web services in the database tier, making SQL Server an HTTP listener. This provides a new type of data access capability for applications that are centralized around Web services. In SQL Server 2005, you can use HTTP to access SQL Server directly without using a middle-tier listener such as Microsoft Internet Information Services (IIS). SQL Server exposes a Web service interface to allow the execution of SQL statements and invocation of functions and procedures. Query results are returned in XML format and can take advantage of the Web services infrastructure of Visual Studio.

XML for Analysis Services (XML/A)

XML for Analysis Services (XML/A) is the native, standards-based protocol for communicating with the Analysis Services server. New kinds of applications are enabled and easy to develop—applications that integrate analytics with operations in real time. With XML/A as the native protocol, Analysis Services clients can be configured to have a zero footprint, and each server is automatically a Web service. A light-footprint Win32 layer is available for backward compatibility with tools that work with Analysis Services 2000 on OLE DB for OLAP, ADOMD, and ADOMD.NET. Many users will continue to use the ADOMD.NET object model to build custom applications on Analysis Services.

Application Framework

SQL Server 2005 introduces a new SQL Server application framework including: Service Broker, Notification Services, SQL Server Mobile, and SQL Server Express. Service Broker is a distributed application framework that provides reliable asynchronous messaging at the database to database level.

Service Broker

Over the last 10 years, the proliferation of e-commerce applications has created the need for increased workflow management across database applications. When an online customer places an order for a book, this order needs to commit transactions into the inventory, shipping, and credit card systems, and also needs to send an order confirmation using another Web application. Waiting for each of these processes to happen in order doesn't scale well. SQL Server 2005 provides a new scalable architecture for building asynchronous message routing. Figure 5 outlines the Service Broker architecture.

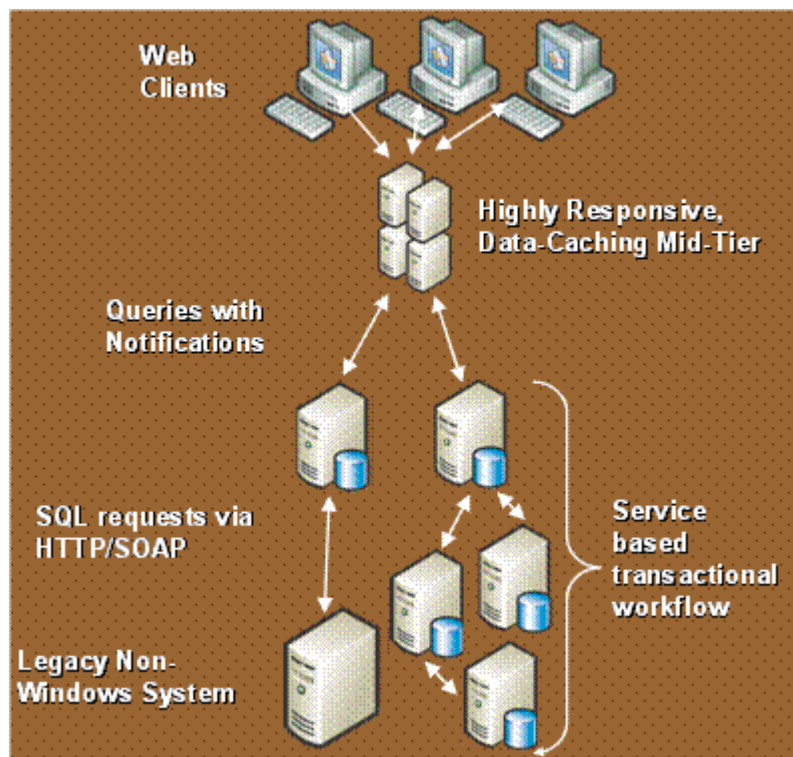


Figure 5: Service Broker architecture

The Service Broker technology allows internal or external processes to send and receive streams of reliable, asynchronous messages by using extensions to normal Transact-SQL data manipulation language. Messages are sent to a queue in the same database as the sender, to another database in the same instance of SQL Server, or to another instance of SQL Server either on the same server or on a remote server.

“SQL Service Broker in SQL Server 2005 has made development 60 percent faster compared to writing SQL jobs.”

--Ketan Patel, Senior Application Developer, Information Technology Corporate Strategy Planning and Analysis, Microsoft

Notification Services

Microsoft SQL Server Notification Services is a platform for developing and deploying applications that generate and send notifications to users. Notifications are personalized, timely messages that can be sent to a wide variety of devices.

Notifications reflect the preferences of the subscriber. The subscriber enters a subscription to express an interest in information. For example, "notify me when the stock price of Adventure Works reaches \$70.00," or "notify me when the strategy document my team is writing is updated."

A notification can be generated and sent to the user as soon as a triggering event occurs, or it can be generated and sent on a predetermined schedule that the user specifies. The user's subscription specifies when the notification should be generated and sent.

Notifications can be sent to a wide variety of devices. For example, a notification can be sent to a user's mobile phone, personal digital assistant (PDA), Microsoft Windows Messenger, or e-mail account. Because these devices often accompany the user, notifications are ideal for sending high-priority information.

SQL Server Mobile Edition

SQL Server 2000 shipped with SQL Server 2000 Windows CE Edition, which is now SQL Server Mobile Edition version 3.0. There are a number of new key features in SQL Server Mobile Edition that relate to developers:

- You can create a SQL Server Mobile Edition database on the desktop or on the device, directly from SQL Server Management Studio. You can also manipulate the schema of the SQL Server Mobile Edition database directly from Management Studio, regardless of whether the database resides on the mobile device or on the desktop. You can use SQL Server Management Studio to run queries that target a SQL Server Mobile Edition database on the device or on the desktop. You can also take advantage of new SQL Server Mobile Edition features that include an XML showplan rendered in a GUI format just like native SQL Server and the ability to use query hints to override the query optimizer in SQL Server Mobile Edition. For the first time, you can control the optimization plan on a device.
- You can now code against SQL Server Integration Services (SSIS) objects to exchange data.
- The new SqlCeResult set is derived from the SQLResult set that is in SQL Server 2005. This allows SQL Server Mobile Edition to have a true scrollable, updateable cursor. It also allows binding to data objects that are on devices.
- You can code an application to synchronize data while leaving the main application open, and you can have two separate applications access the same database on the device at the same time.
- You can get notifications that you can code into status bars that will give the status of a synchronization. Previously, there was no way to know how far synchronization status was, to notify users that a device had not stopped responding.
- You can maintain the small size of the database through a much more aggressive page reclamation policy.
- You can share parameterized query code with SQL Server syntax.

SQL Server Express

More than ever developers are leveraging relational databases to provide a rich end-user experience. Protecting and managing information inside these applications is critical. Microsoft SQL Server Express helps developers build robust and reliable applications by providing a free, easy to use, and robust database. Too often database systems are overly complex for building simple applications. Microsoft Visual Studio 2005 and SQL Server Express reduce this complexity by providing a simple but powerful development environment for building data-driven applications. Developers can design schemas, add data, and query local databases, all inside the Visual Studio 2005 environment. If developers need more advanced database features, then SQL Server Express can be seamlessly upgraded to more sophisticated versions of SQL Server. Figure 6 shows the Query Editor interface in SQL Server Express Manager

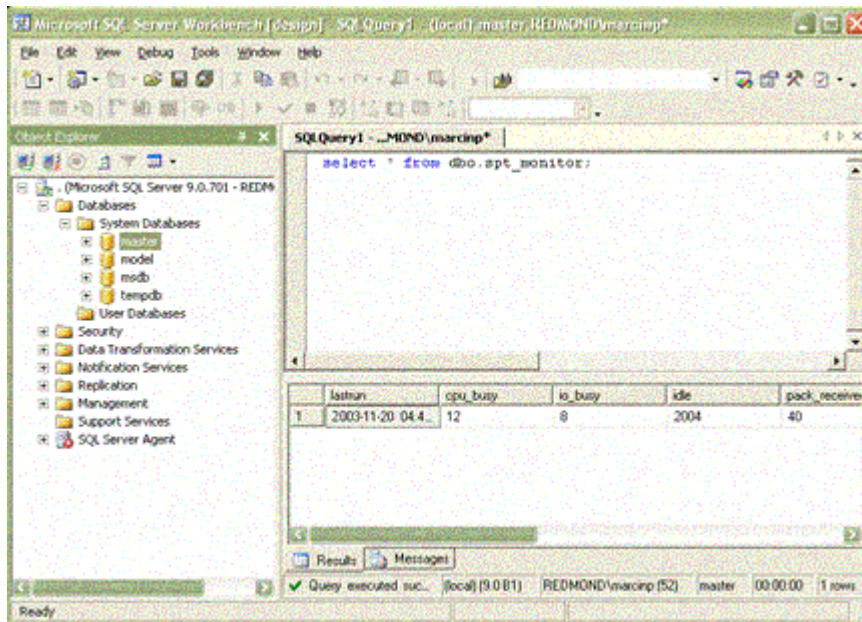


Figure 6: The Query Editor in SQL Server Express Manager (XM)

A new GUI tool called SQL Server Express Manager (XM) is freely available as a separate Web download. XM allows easy database management and query analysis capabilities, will have a small download size, and will be freely redistributable. XM supports connections to SQL Server Express and other SQL Server 2005 editions, SQL Server 2000, and MSDE 2000. A simplified connection dialog box guides the user through the selection of the instance and the authentication methods to be used. Both local and remote connections are possible using XM. Object Explorer will enumerate and display the common objects used, such as the instance, tables, stored process, etc., in a hierarchical manner and will help the user visualize access to the database.

All database management functionalities are available by invoking the right-click context menu from Object Explorer. Some of the database management options to be exposed include creating and modifying databases, tables, logins, and users. Many of these common database operations are available as task wizards that guide the user through the process, while many others are available as tabbed window documents. For instance, XM will provide a New/Edit Database document for creating new databases and editing existing databases.

Many database users prefer to manage their servers using Transact-SQL, since this approach offers finer-grained control than using the graphical user interface. The Query Editor in XM will allow users to develop and execute Transact-SQL statements and scripts. The Query Editor will have rich features such as keyword color-coding and a results pane that returns results in a data grid. The error messages, if any, will also be shown in the results pane.

Business Intelligence

SQL Server 2005 will further Microsoft's leadership in the area of business intelligence (BI) through innovations in scalability, data integration, development tools, and rich analytics. SQL Server 2005 enables scalable business intelligence by putting critical, timely information in the hands of employees across your organization. From the CEO to the information worker, employees will be able to quickly and easily harness data to make better decisions faster. The comprehensive integration, analysis, and reporting capabilities of SQL Server 2005 enable companies to extend the value of their existing applications, regardless of the underlying platform.

Business intelligence features include enhancements in the following areas:

- End-to-end integrated business intelligence platform
- Integration Services
- Analysis Services
- Reporting Services
- Integration with the Microsoft Office System

End-to-End Integrated Business Intelligence Platform

SQL Server 2005 is a complete business intelligence platform that provides the features, tools, and functionality to build both classic and innovative kinds of analytical applications. The following provides an introduction to the tools that you will use to build an analytical application, and highlights new functionality that makes it easier than ever to build and manage complex BI systems.

The SQL Server 2005 business intelligence toolset delivers end-to-end BI application integration:

- **Design:** Business Intelligence Development Studio is the first integrated development environment designed for the business intelligence developer. Built on Visual Studio 2005, the Business Intelligence Development Studio delivers a rich, integrated, professional development platform for BI system developers. Debugging, source control, and script and code development are available for all components of the BI platform.
- **Integrate:** SQL Server Integration Services (SSIS) has been rewritten to perform complex data integration, transformation, and synthesis at high speed for very large data volumes. The Business Intelligence Development Studio makes building and debugging packages positively fun. Integration Services, Analysis Services, and Reporting Services work together to present a seamless view of data from heterogeneous sources.

-
- **Analyze:** Microsoft Data Mining has always been easy to use. Now it's even better with the addition of important new algorithms, including Association Rules, Time Series, Regression Trees, Sequence Clustering, Neural Network, and Naïve Bayes. SQL Server 2005 blurs the lines between relational and multidimensional databases. You can store data in the relational database, in the multidimensional database, or use the new Proactive Cache feature to get the best of both worlds. Important new analytical capabilities have been added to Analysis Services cubes as well: these include Key Performance Indicator (KPI) framework, MDX scripts, and other built-in advanced business analytics. The Reporting Services report delivery and management framework enables easy distribution of complex analytics to the widest possible audience.
 - **Report:** Reporting Services extends the Microsoft business intelligence platform to reach the business user who needs to consume the analysis. Reporting Services is an enterprise managed reporting environment, embedded and managed via Web services. Reports can be personalized and delivered in a variety of formats, with a range of interactivity and printing options. Complex analyses can reach a broad audience through the distribution of reports as a data source for downstream business intelligence. New with SQL Server 2005 is the reporting tool, Report Builder.
 - **Manage:** SQL Server Management Studio integrates the management of all SQL Server 2005 components. Business intelligence practitioners will benefit from Microsoft's extension of the server abilities you expect from the relational engine—scalability, reliability, availability, programmability, and so on—to the full set of BI platform components.

Integration Services

SQL Server 2005 includes a redesigned enterprise ETL platform, called SQL Server Integration Services (SSIS). SQL Server Integration Services enables organizations to more easily integrate and analyze data from multiple heterogeneous information sources. By analyzing data across a wide array of operational systems, organizations may gain a competitive edge through a holistic understanding of their business.

Enterprise ETL Platform

This new platform is the successor to the popular feature in SQL Server 2000, called Data Transformation Services (DTS). SSIS is completely new for SQL Server 2005. SSIS provides the breadth of features, and very high scale performance that is necessary to build enterprise-class ETL applications. SSIS is fully programmable, embeddable, and extensible—characteristics that make it an ideal ETL platform.

Beyond Traditional ETL

SQL Server 2005 supports nontraditional data (Web Services, XML) out-of-the box through:

- SSIS brings analytics to the data without persisting the data.
- Data Mining and text mining in the data flow.
- Data Mining and analytics are brought to the data flow for data quality and data cleansing.

Analysis Services

With SQL Server 2005, Analysis Services provides, for the first time, a unified and integrated view of all your business data as the foundation for all of your traditional reporting, OLAP analysis, and data mining.

“The move to Analysis Services in SQL Server 2005 has been a huge benefit for our operations. ...We’re seeing at least a 75 percent reduction in OLAP cube processing time.”

--Mike McGrew, Development Manager, Information Technology Tax and Audit Product Group, Microsoft

Unified Dimensional Model

By combining the best aspects of traditional OLAP analysis and relational reporting, Analysis Services 2005 provides a metadata model that covers both sets of needs. A set of cubes and dimensions defined in Analysis Services 2005 is referred to as a Unified Dimensional Model (UDM). The UDM is a central metadata repository defining business entities, business logic, calculations, and metrics that serves as the source for all reports, spreadsheets, OLAP browsers, KPIs, and analytical applications.

Using the powerful new Data Source View feature, the UDM is mapped to a host of heterogeneous back-end data sources allowing a complete and integrated picture of the business regardless of the location of the data.

With the UDM’s friendly descriptions of the business entities, navigation hierarchies, multiple perspectives, and even automatic translations to native languages, users will find it easy to explore the corporate business data.

Data Mining

Microsoft SQL Server 2005 Data Mining is the business intelligence technology that helps you build complex analytical models, and integrate those models with your business operations. Microsoft SQL Server 2005 Analysis Services establishes new ground for data mining. By creating an easy to use, extensible, accessible, and flexible platform, SQL Server 2005 Analysis Services data mining capabilities introduces data mining to organizations that previously would never have considered a data mining solution.

Through an enterprise class architecture; a deep integration with the SQL Server family of business intelligence tools, and a rich set of tools, APIs and algorithms, SQL Server enables the creation of a new breed of intelligent applications that enhance productivity, increase profits and reduce costs by providing customized data-driven solutions to a broad range of business problems.

Reporting Services

Reporting Services extends the Microsoft BI platform to reach the information worker who needs access to business data. Reporting Services is a server-based enterprise reporting environment, managed via Web services. Reports can be delivered in a variety of formats, with a range of interactivity and printing options. Complex analyses can reach a broad audience through the distribution of reports as a data source for downstream business intelligence.

An integrated component of SQL Server 2005, Reporting Services provides:

- A high performance engine for processing and formatting reports.
- A complete set of tools for creating, managing, and viewing reports.
- An extensible architecture and open interfaces for embedding reports or integrating reporting solutions in diverse Information Technology environments.

Relational and OLAP Reports

Reports built on relational data are useful but the ability to add additional analytic capabilities makes reporting powerful. Reporting Services allows you to easily build reports together or separately. SQL Server 2005 supports both relational and OLAP data and provides a query editor for both including SQL Query Editor and MDX Query Editor.

Report Builder

Report Builder, a new component of SQL Server 2005 Reporting Services, allows business users to create their own reports using a user-friendly model of their data. Report Builder leverages the Reporting Services platform to bring ad hoc reporting to all end users. Users create and edit reports with the Report Builder client application. The Report Builder user interface is built on top of familiar Microsoft Office paradigms such as Excel and PowerPoint. Figure 7 shows a sample Report Builder report.

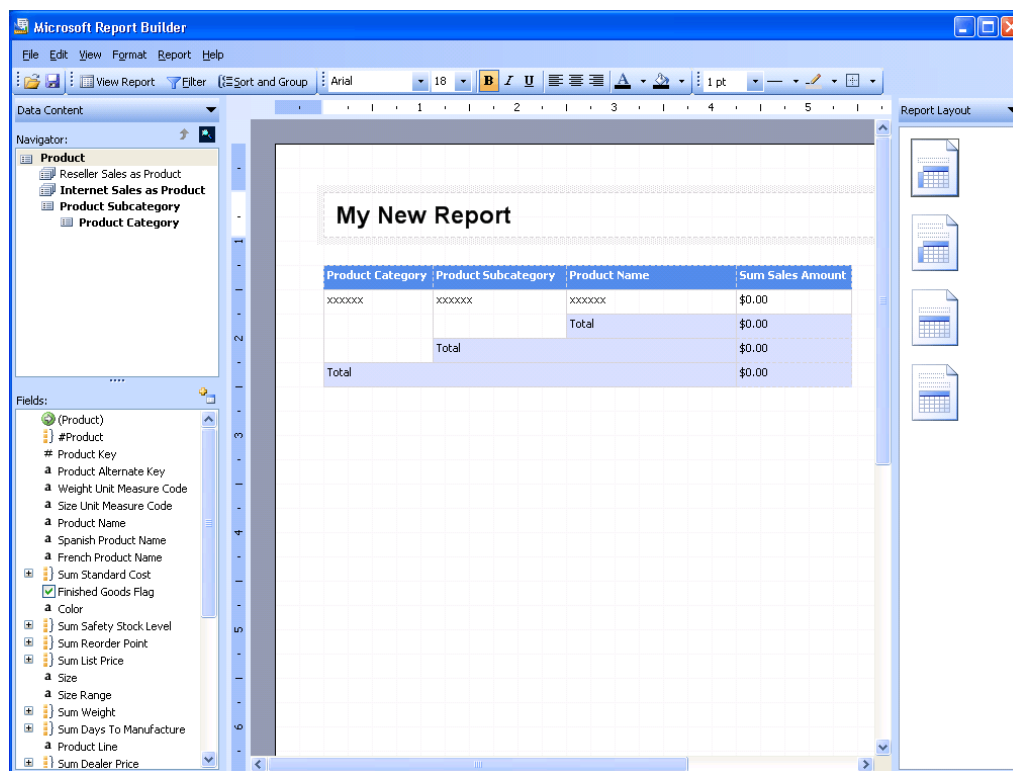


Figure 7: Design reports with Report Builder

Report Builder is a ClickOnce application deployed via the browser. Users start by selecting report layout templates containing predefined data sections such as tables, matrices and charts. They drag and drop report items from the model to the design surface and set constraints to filter the report data. The model contains all of the necessary information for the Report Builder to automatically generate the source query and retrieve the requested data. The Report Builder also allows users to:

- Add text and formatting to reports.
- Create new fields and calculations defined using the model.
- Preview, print, and publish reports.
- Export report data to formats such as Microsoft Excel.

Integration with the Microsoft Office System

Reports that are served up by the Report Server in Reporting Services can run in the context of Microsoft SharePoint® Portal Server and Microsoft Office System applications such as Microsoft Word and Microsoft Excel. You can use SharePoint features to subscribe to reports, create new versions of reports, and distribute reports. You can also open reports in Word or Excel to view HTML versions of the reports.

Upgrading to SQL Server 2005

Following are some tips for upgrading to SQL Server 2005.

- Upgrade to SQL Server 2005 from SQL Server 7.0 or SQL Server 2000.
- Run Upgrade Advisor before upgrading to determine if any product changes are expected to break existing applications.
- The Database Engine, Analysis Services, and Reporting Services can be upgraded by Setup.
- SQL Server Integration Services, the replacement for DTS, is installed side-by-side with DTS. You can run DTS packages using the DTS runtime components.
- SQL Server 2005 Notification Services is installed side-by-side with Notification Services 2.0. You must migrate instances of Notification Services to SQL Server 2005 when you upgrade the Database Engine.
- Use the Surface Area Configuration tool after upgrading to enable or disable SQL Server 2005 services, network protocols, and features.

Microsoft SQL Server Pricing and Licensing

The family of SQL Server editions includes everything you need in one product for a comprehensive, out-of-the-box data storage, management, analysis and reporting platform with flexible licensing that allows you to choose the solution that best fits your unique needs.

Designed to scale from the largest enterprise down to the smallest business, SQL Server provides the same performance, security, reliability, and business value to all customers. SQL Server supports implementations ranging from multi-terabyte data warehouses to Pocket PC devices running SQL Server Windows CE Edition.

Pricing & licensing

Edition	Pricing	Benefit	Size	Key Features
Express	Free	Fastest way to learn, build and deploy simple data-driven applications.	1 CPU 1-GB RAM 4-GB DB size	Simple Management Tool Simple Reporting Replication & SSB Client
Workgroup	\$3,900 per processor \$739 (server + 5 users)	Most affordable and easiest to use database solution for smaller departments and growing businesses.	1-2 CPUs 3-GB RAM	Management Studio Import/Export Limited Replication Publishing Clustering Back-up Log Shipping
Standard	\$6,000 per processor \$2,799 (server + 10 users)	Complete data management and analysis platform for medium businesses and larger departments.	1-4 CPUs Unlimited RAM	Database Mirroring Basic ETL Standard OLAP Server with Analysis Services Standard Reporting with Reporting Services Data Mining Full Replication & SSB Publishing Available in native 32- and 64-bit editions Supports Itanium2 and x64
Enterprise	\$25,000 per processor \$13,500 (server + 25 users)	Fully integrated data management and analysis platform for business-critical enterprise applications.	Unlimited Scale & Partitioning	Advanced database mirroring, Complete online & parallel operations, and database snapshot. Advanced analysis tools including full OLAP & Data Mining. Advanced reporting with customized, high scale, and ad hoc reporting.

Edition	Pricing	Benefit	Size	Key Features
				Advanced ETL with complex data routing and transformation capabilities. Available in native 32 and 64 bit editions. Supports Itanium2 and x64

This table outlines the pricing and benefits for each edition of the Microsoft SQL Server 2005 product line.

Note **Bold** indicates a feature that is new for Microsoft SQL Server 2005. Each higher edition includes the same functionality as the edition below it. All prices are in US dollars and reflect pricing for purchases within the United States.

For more information on pricing and licensing and the SQL Server 2005 product line please visit the following:

<http://www.microsoft.com/sql/howtobuy/default.asp>

<http://www.microsoft.com/sql/howtobuy/understdbpricing.asp>

Windows Server System Common Engineering Roadmap

SQL Server is part of the Windows Server System—a comprehensive and integrated server infrastructure that simplifies the development, deployment, and operation of flexible business solutions.

As a key part of the Windows Server System family, SQL Server 2005 includes a standard set of capabilities such as common patch management, Watson support, and tools such as the Microsoft Baseline Security Analyzer to deliver a consistent and predictable experience to Windows Server System customers.

The goal of the Windows Server System Common Engineering Roadmap is to create a set of common services that will be implemented across all Windows Server System server products. This set of common services raises the bar for server infrastructure and helps ensure that products throughout Windows Server System are engineered for greater security, reliability, manageability, and flexibility. The Windows Server System Common Engineering Roadmap builds on the following initiatives to enable customers to meet today's business challenges and the challenges of tomorrow.

To deliver on this vision of flexible Information Technology, Microsoft is currently focusing on three key initiatives:

- **.NET**, Microsoft's Web services strategy, connects information, people, systems, and devices through software. Adopting services-oriented architecture throughout the Microsoft platform provides businesses with the ability to quickly build, deploy, manage, and use connected, security-enhanced systems based on Web services. These systems enable faster, more agile business integration and deliver on the promise of information anytime, anywhere, on any device.
- **Dynamic Systems Initiative (DSI)** is focused on delivering systems that are designed with operations in mind and are built to monitor ongoing operations and adjust dynamically based on models that can change with the business. This initiative unifies hardware, software, and service vendors around a model-based management that enables customers to harness the power of industry-standard hardware and brings simplicity, automation, and flexibility to Information Technology operations. The goal is to reduce management and operations costs, improve reliability, and increase responsiveness throughout the entire Information Technology life cycle.
- **Trustworthy Computing** is a long-term, company-wide, and collaborative effort to create and deliver more secure, private, and reliable computing experiences for users, while reducing the demands on users and Information Technology administrators. The goal of Trustworthy Computing is to deliver the security, privacy, reliability, and business integrity that people expect from the computing industry.

In the Common Engineering Criteria for 2005, 16 different specifications have been defined and applied throughout the Windows Server System. Beginning with 2005 versions, all Windows Server System products will either comply with these criteria, or have specific reasons for any exemptions, with implementation plans for future releases.

Microsoft will add specifications at regular intervals to continue to address customer requirements and deliver on the vision for Windows Server System. These specifications and their implementation details, including any exception information, will be published regularly on the Windows Server System Web site, <http://www.microsoft.com/windowsserversystem>.

Conclusion

SQL Server 2005 provides the technology and capabilities that organizations count on. With significant advancements in the key areas of enterprise data management, developer productivity, and business intelligence, the benefits of SQL Server 2005 are substantial.

This paper demonstrates that SQL Server 2005 can benefit your organization in the following ways:

- **Leverage data assets:** In addition to delivering a secure, reliable database for line-of-business and analytical applications, SQL Server 2005 enables customers to get more value from their data by including embedded functionality such as reporting, analysis, and data mining.
- **Increase productivity:** Through comprehensive business intelligence capabilities and integration with familiar tools like Office, SQL Server 2005 provides information workers in your organization with critical, timely business information that is tailored to their specific needs. Our goal is to extend BI to all users within an organization and ultimately to allow users at all levels of the organization to make better business decisions based on one of their most valuable assets—their data.
- **Reduce Information Technology complexity:** SQL Server 2005 simplifies the development, deployment, and management of line-of-business and analytical applications by providing a flexible development environment for developers and integrated, automated management tools for database administrators.
- **Lower Total Cost of Ownership (TCO):** Our integrated approach and focus on ease-of-use provides the industry's lowest upfront, implementation, and maintenance costs for rapid returns on your database investment.

This paper provides only an overview of all the new benefits and functionality in SQL Server 2005. For more information, please visit the following:

SQL Server site on TechNet: <http://www.microsoft.com/technet/prodtechnol/sql>

SQL Server Developer Center: <http://msdn.microsoft.com/sql>

SQL Server 2005 Web site: <http://www.microsoft.com/sql/2005>