



# Visited 01/22/2016

SQL Server 2008 (formerly codenamed "Katmai")<sup>[12][13]</sup> was released on August 6, 2008, announced to the SQL Server Special Interest Group at the ESRI 2008 User's Conference on August 6, 2008 by Ed Katibah (Spatial Program Manager at Microsoft), and aims to make data management *self-tuning*, self organizing, and self maintaining with the development of *SQL Server Always On* technologies, to provide near-zero downtime. SQL Server 2008 also includes support for *structured* and semi-structured data, including digital media formats for pictures, audio, video and other multimedia data. In current versions, such multimedia data can be stored as *BLOBs* (binary large objects), but they are generic bitstreams. Intrinsic awareness of multimedia data will allow specialized functions to be performed on them. According to *Paul Flessner*, senior Vice President of Server Applications at Microsoft, SQL Server 2008 can be a data storage backend for *different varieties of data: XML, email, time/calendar, file, document, spatial, etc.* as well as perform *search, query, analysis, sharing, and synchronization* across all data types.<sup>[13]</sup>

Other new data types include specialized date and time types and a *Spatial* data type for location-dependent data.<sup>[14]</sup> Better support for unstructured and semi-structured data is provided using the new *FILESTREAM*<sup>[15]</sup> data type, which can be used to reference any file stored on the file system.<sup>[16]</sup> Structured data and metadata about the file is stored in SQL Server database, whereas the unstructured component is stored in the file system. Such files can be accessed both via *Win32* file handling *APIs* as well as via SQL Server using *T-SQL*; doing the latter accesses the file data as a *BLOB*. Backing up and restoring the database backs up or restores the referenced files as well.<sup>[17]</sup> SQL Server 2008 also natively supports hierarchical data, and includes *T-SQL* constructs to directly deal with them, without using recursive queries.<sup>[17]</sup>

The *full-text search* functionality has been integrated with the database engine. According to a Microsoft technical article, this simplifies management and improves performance.<sup>[18]</sup>

Spatial data will be stored in two types. A "Flat Earth" (GEOMETRY or planar) data type represents geospatial data which has been projected from its native, spherical, coordinate system into a plane. A "Round Earth" data type (GEOGRAPHY) uses an ellipsoidal model in which the Earth is defined as a single continuous entity which does not suffer from the singularities such as the international dateline, poles, or map projection zone "edges". Approximately 70 methods are available to represent spatial operations for the Open Geospatial Consortium *Simple Features for SQL*, Version 1.1.<sup>[19]</sup>

SQL Server includes better compression features, which also helps in improving scalability.<sup>[20]</sup> It enhanced the indexing algorithms and introduced the notion of filtered indexes. It also includes *Resource Governor* that allows reserving resources for certain users or workflows. It also includes capabilities for *transparent encryption of data* (TDE) as well as compression of backups.<sup>[19]</sup> SQL Server 2008 supports the *ADO.NET Entity Framework* and the reporting tools, replication, and data definition will be built around the *Entity Data Model*.<sup>[21]</sup> *SQL Server Reporting Services* will gain charting capabilities from the integration of the data visualization products from *Dundas Data Visualization, Inc.*, which was acquired by Microsoft.<sup>[22]</sup> On the management side, SQL Server 2008 includes the *Declarative Management Framework* which allows configuring policies and constraints, on the entire database or certain tables, declaratively.<sup>[14]</sup> The version of *SQL Server Management Studio* included with SQL Server 2008 supports *IntelliSense* for SQL queries against a SQL Server 2008 Database Engine.<sup>[23]</sup> SQL Server 2008 also makes the databases available via *Windows PowerShell* providers and management functionality available as *Cmdlets*, so that the server and all the running instances can be managed from *Windows PowerShell*.<sup>[24]</sup>

The final SQL Server 2008 service pack (10.00.6000, Service Pack 4) was released on September 30, 2014.<sup>[25]</sup>

## SQL Server 2008 R2 [ edit ]

SQL Server 2008 R2 (10.50.1600.1, formerly codenamed "Kilimanjaro") was announced at TechEd 2009, and was released to manufacturing on April 21, 2010.<sup>[26]</sup> SQL Server 2008 R2 adds certain features to SQL Server 2008 including a *master data management* system branded as *Master Data Services*, a central management of master data entities and hierarchies. Also Multi Server Management, a centralized console to manage multiple SQL Server 2008 instances and services including relational databases, Reporting Services, Analysis Services & Integration Services.<sup>[27]</sup>

SQL Server 2008 R2 includes a number of new services,<sup>[28]</sup> including *PowerPivot* for *Excel* and *SharePoint*, *Master Data Services*, *StreamInsight*, *Report Builder 3.0*, *Reporting Services* Add-in for *SharePoint*, a Data-tier function in *Visual Studio* that enables packaging of tiered databases as part of an application, and a SQL Server Utility named UC (Utility Control Point), part of *AMSM* (Application and Multi-Server Management) that is used to manage multiple SQL Servers.<sup>[29]</sup>

The first SQL Server 2008 R2 service pack (10.50.2500, Service Pack 1) was released on July 11, 2011.<sup>[30]</sup>

The second SQL Server 2008 R2 service pack (10.50.4000, Service Pack 2) was released on July 26, 2012.<sup>[31]</sup>

The final SQL Server 2008 R2 service pack (10.50.6000, Service Pack 3) was released on September 26, 2014.<sup>[32]</sup>

## SQL Server 2012 [ edit ]

At the 2011 *Professional Association for SQL Server* (PASS) summit on October 11, Microsoft announced that the next major version of SQL Server (codenamed "Denali"), would be SQL Server 2012. It was released to manufacturing on March 6, 2012.<sup>[33]</sup> SQL Server 2012 Service Pack 1 was released to manufacturing on November 9, 2012, and Service Pack 2 was released to manufacturing on June 10, 2014.

It was announced to be the last version to natively support *OLE DB* and instead to prefer *ODBC* for native connectivity.<sup>[34]</sup>

SQL Server 2012's new features and enhancements include *Always On SQL Server Failover Cluster Instances* and *Availability Groups* which provides a set of options to improve database availability.<sup>[35]</sup> *Contained Databases* which simplify the moving of databases between instances, new and modified *Dynamic Management Views* and *Functions*,<sup>[36]</sup> programmability enhancements including new spatial features,<sup>[37]</sup> metadata discovery, sequence objects and the *THROW* statement,<sup>[38]</sup> performance enhancements such as *ColumnStore Indexes* as well as improvements to *OnLine* and partition level operations and security enhancements including provisioning during setup, new permissions, improved role management, and default schema assignment for groups.<sup>[39][40]</sup>

## SQL Server 2014 [ edit ]

SQL Server 2014 was released to manufacturing on March 18, 2014, and released to the general public on April 1, 2014. Until November 2013 there were two CTP revisions, CTP1 and CTP2.<sup>[41]</sup> SQL Server 2014 provides a new in-memory capability for tables that can fit entirely in memory (also known as *Hekaton*). Whilst small tables may be entirely resident in memory in all versions of SQL Server, they also may reside on disk, so work is involved in reserving RAM, writing evicted pages to disk, loading new pages from disk, locking the pages in RAM while they are being operated on, and many other tasks. By treating a table as guaranteed to be entirely resident in memory much of the "plumbing" of disk-based databases can be avoided.<sup>[42]</sup>

For disk-based SQL Server applications, it also provides the *SSD Buffer Pool Extension*, which can improve performance by cache between *DRAM* and spinning media.

SQL Server 2014 also enhances the *Always On* (HADR) solution by increasing the readable secondaries count and sustaining read operations upon secondary-primary disconnections, and it provides new hybrid disaster recovery and backup solutions with *Microsoft Azure*, enabling customers to use existing skills with the on-premises version of SQL Server to take advantage of Microsoft's global datacenters. In addition, it takes advantage of new *Windows Server 2012* and *Windows Server 2012 R2* capabilities for database application scalability in a physical or virtual environment.

Microsoft provides three versions of SQL Server 2014 for downloading: the one that runs on *Microsoft Azure*, the SQL Server 2014 CAB, and SQL Server 2014 ISO.<sup>[43]</sup>

SQL Server 2014 SP1, consisting primarily of bugfixes, was released on May 15, 2015.<sup>[44]</sup>

## SQL Server 2016 [ edit ]

Microsoft SQL Server 2016 Community Technology Preview 3.2 (CTP 3.2) is the most recent pre-release version available. It was released on December 7, 2015.<sup>[45]</sup>

## Editions [ edit ]

Microsoft makes SQL Server available in multiple editions, with different feature sets and targeting different users. These editions are:<sup>[46][47]</sup>

### Mainstream editions [ edit ]

#### Datacenter

SQL Server 2008 R2 Datacenter is a full-featured edition of SQL Server and is designed for datacenters that need high levels of application support and scalability. It supports 256 logical processors and virtually unlimited memory and comes with *StreamInsight Premium* edition.<sup>[48]</sup> The Datacenter edition has been retired in SQL Server 2012; all of its features are available in SQL Server 2012 Enterprise Edition.<sup>[49]</sup>

#### Enterprise

SQL Server Enterprise Edition includes both the core database engine and add-on services, with a range of tools for creating and managing a SQL Server cluster. It can manage databases as large as 524 *petabytes* and address 2 *terabytes* of memory and supports 8 physical processors. SQL Server 2012 Enterprise Edition supports 160 physical processors.<sup>[50]</sup>

#### Standard

SQL Server Standard edition includes the core database engine, along with the stand-alone services. It differs from Enterprise edition in that it supports fewer active instances (number of nodes in a cluster) and does not include some high-availability functions such as hot-add memory (allowing memory to be added while the server is still running), and parallel indexes.

#### Web

SQL Server Web Edition is a low-*TCO* option for Web hosting.

#### Business Intelligence

Introduced in SQL Server 2012 and focusing on Self Service and Corporate Business Intelligence. It includes the Standard Edition capabilities and Business Intelligence tools: *PowerPivot*, *Power View*, the *BI Semantic Model*, *Master Data Services*, *Data Quality Services* and *xVelocity in-memory analytics*.<sup>[51]</sup>

#### Workgroup

SQL Server Workgroup Edition includes the core database functionality but does not include the additional services. Note that this edition has been retired in SQL Server 2012.<sup>[49]</sup>

#### Express

SQL Server Express Edition is a scaled down, free edition of SQL Server, which includes the core database engine. While there are no limitations on the number of databases or users supported, it is limited to using one processor, 1 GB memory and 10 GB database files (4 GB database files prior to SQL Server Express 2008 R2).<sup>[52]</sup> It is intended as a replacement for *MSDE*. Two additional editions provide a superset of features not in the original Express Edition. The first is *SQL Server Express with Tools*, which includes *SQL Server Management Studio Basic*. *SQL Server Express with Advanced Services* adds full-text search capability and reporting services.<sup>[53]</sup>

### Specialized editions [ edit ]

#### Azure

Azure SQL Database is the *cloud-based* version of Microsoft SQL Server, presented as a *platform as a service* offering on *Microsoft Azure*.

#### Compact (SQL CE)

The compact edition is an embedded database engine. Unlike the other editions of SQL Server, the SQL CE engine is based on *SQL Mobile* (initially designed for use with hand-held devices) and does not share the same binaries. Due to its small size (1 MB DLL footprint), it has a markedly reduced feature set compared to the other editions. For example, it supports a subset of the standard data types, does not support stored procedures or *Views* or multiple-statement batches (among other limitations). It is limited to 4 GB maximum database size and cannot be run as a *Windows* service, Compact Edition must be hosted by the application using it. The 3.5 version includes support for *ADO.NET Synchronization Services*. SQL CE does not support *ODBC* connectivity, unlike SQL Server proper.

## Developer

SQL Server Developer Edition includes the same features as SQL Server 2012 Enterprise Edition, but is limited by the license to be only used as a development and test system, and not as production server. This edition is available to download by students free of charge as a part of [Microsoft's DreamSpark program](#).<sup>[54]</sup>

## Embedded (SSEE)

SQL Server 2005 Embedded Edition is a specially configured named instance of the SQL Server Express database engine which can be accessed only by certain Windows Services.

## Evaluation

SQL Server Evaluation Edition, also known as the *Trial Edition*, has all the features of the Enterprise Edition, but is limited to 180 days, after which the tools will continue to run, but the server services will stop.<sup>[55]</sup>

## Fast Track

SQL Server Fast Track is specifically for enterprise-scale data warehousing storage and business intelligence processing, and runs on reference-architecture hardware that is optimized for Fast Track.<sup>[56]</sup>

## LocalDB

Introduced in SQL Server Express 2012, LocalDB is a minimal, on-demand, version of SQL Server that is designed for application developers.<sup>[57]</sup> It can also be used as an embedded database.<sup>[58]</sup>

## Analytics Platform System (APS)

Formerly Parallel Data Warehouse (PDW) A [massively parallel processing](#) (MPP) SQL Server appliance optimized for large-scale [data warehousing](#) such as hundreds of terabytes.<sup>[59]</sup>

## Datawarehouse Appliance Edition

Pre-installed and configured as part of an appliance in partnership with Dell & HP base on the Fast Track architecture. This edition does not include SQL Server Integration Services, Analysis Services, or Reporting Services.

## Architecture [\[edit\]](#)

The protocol layer implements the external interface to SQL Server. All operations that can be invoked on SQL Server are communicated to it via a Microsoft-defined format, called [Tabular Data Stream](#) (TDS). TDS is an application layer protocol, used to transfer data between a database server and a client. Initially designed and developed by Sybase Inc. for their [Sybase SQL Server](#) relational database engine in 1984, and later by Microsoft in Microsoft SQL Server, TDS packets can be encased in other physical transport dependent protocols, including [TCP/IP](#), [named pipes](#), and [shared memory](#). Consequently, access to SQL Server is available over these protocols. In addition, the SQL Server API is also exposed over [web services](#).<sup>[47]</sup>

## Data storage [\[edit\]](#)

[Data storage](#) is a [database](#), which is a collection of tables with [typed](#) columns. SQL Server supports different data types, including [primary types](#) such as *Integer*, *Float*, *Decimal*, *Char* (including character strings), *Varchar* (variable length character strings), *binary* (for unstructured [blobs](#) of data), *Text* (for textual data) among others. The [rounding](#) of floats to integers uses either Symmetric Arithmetic Rounding or Symmetric Round Down (*Fx*) depending on arguments: `SELECT Round(2.5, 0)` gives 3.

Microsoft SQL Server also allows user-defined composite types (UDTs) to be defined and used. It also makes server statistics available as virtual tables and views (called Dynamic Management Views or DMVs). In addition to tables, a database can also contain other objects including [views](#), [stored procedures](#), [indexes](#) and [constraints](#), along with a transaction log. A SQL Server database can contain a maximum of 2<sup>31</sup> objects, and can span multiple OS-level files with a maximum file size of 2<sup>60</sup> bytes (1 exabyte).<sup>[47]</sup> The data in the database are stored in primary data files with an extension `.mdf`. Secondary data files, identified with a `.ndf` extension, are used to allow the data of a single database to be spread across more than one file, and optionally across more than one file system. Log files are identified with the `.ldf` extension.<sup>[47]</sup>

Storage space allocated to a database is divided into sequentially numbered *pages*, each 8 KB in size. A *page* is the basic unit of I/O for SQL Server operations. A page is marked with a 96-byte header which stores metadata about the page including the page number, page type, free space on the page and the ID of the object that owns it. Page type defines the data contained in the page: data stored in the database, index, allocation map which holds information about how pages are allocated to tables and indexes, change map which holds information about the changes made to other pages since last backup or logging, or contain large data types such as image or text. While page is the basic unit of an I/O operation, space is actually managed in terms of an *extent* which consists of 8 pages. A database object can either span all 8 pages in an extent ("uniform extent") or share an extent with up to 7 more objects ("mixed extent"). A row in a database table cannot span more than one page, so is limited to 8 KB in size. However, if the data exceeds 8 KB and the row contains *Varchar* or *Varbinary* data, the data in those columns are moved to a new page (or possibly a sequence of pages, called an *Allocation unit*) and replaced with a pointer to the data.<sup>[60]</sup>

For physical storage of a table, its rows are divided into a series of partitions (numbered 1 to n). The partition size is user defined; by default all rows are in a single partition. A table is split into multiple partitions in order to spread a database over a [computer cluster](#). Rows in each partition are stored in either [B-tree](#) or [heap](#) structure. If the table has an associated, clustered [index](#) to allow fast retrieval of rows, the rows are stored in-order according to their index values, with a B-tree providing the index. The data is in the leaf node of the leaves, and other nodes storing the index values for the leaf data reachable from the respective nodes. If the index is non-clustered, the rows are not sorted according to the index keys. An indexed [view](#) has the same storage structure as an indexed table. A table without a clustered index is stored in an unordered heap structure. However, the table may have non-clustered indices to allow fast retrieval of rows. In some situations the heap structure has performance advantages over the clustered structure. Both heaps and B-trees can span multiple allocation units.<sup>[61]</sup>

## Buffer management [\[edit\]](#)

SQL Server [buffers](#) pages in [RAM](#) to minimize disc I/O. Any 8 KB page can be buffered in-memory, and the set of all pages currently buffered is called the buffer cache. The amount of memory available to SQL Server decides how many pages will be cached in memory. The buffer cache is managed by the *Buffer Manager*. Either reading from or writing to any page copies it to the buffer cache. Subsequent reads or writes are redirected to the in-memory copy, rather than the on-disc version. The page is updated on the disc by the Buffer Manager only if the in-memory cache has not been referenced for some time. While writing pages back to disc, [asynchronous I/O](#) is used whereby the I/O operation is done in a background thread so that other operations do not have to wait for the I/O operation to complete. Each page is written along with its [checksum](#) when it is written. When reading the page back, its checksum is computed again and matched with the stored version to ensure the page has not been damaged or tampered with in the meantime.<sup>[62]</sup>

## Concurrency and locking [\[edit\]](#)

SQL Server allows multiple clients to use the same database concurrently. As such, it needs to control concurrent access to shared data, to ensure data integrity—when multiple clients update the same data, or clients attempt to read data that is in the process of being changed by another client. SQL Server provides two modes of concurrency control: [pessimistic concurrency](#) and [optimistic concurrency](#). When pessimistic concurrency control is being used, SQL Server controls concurrent access by using locks. Locks can be either shared or exclusive. Exclusive lock grants the user exclusive access to the data—no other user can access the data as long as the lock is held. Shared locks are used when some data is being read—multiple users can read from data locked with a shared lock, but not acquire an exclusive lock. The latter would have to wait for all shared locks to be released. Locks can be applied on different levels of granularity—on entire tables, pages, or even on a per-row basis on tables. For indexes, it can either be on the entire index or on index leaves. The level of granularity to be used is defined on a per-database basis by the database administrator. While a fine grained locking system allows more users to use the table or index simultaneously, it requires more resources. So it does not automatically turn into higher performing solution. SQL Server also includes two more lightweight [mutual exclusion](#) solutions—latches and spinlocks—which are less robust than locks but are less resource intensive. SQL Server uses them for DMVs and other resources that are usually not busy. SQL Server also monitors all worker threads that acquire locks to ensure that they do not end up in [deadlocks](#)—in case they do, SQL Server takes remedial measures, which in many cases is to kill one of the threads entangled in a deadlock and rollback the transaction it started.<sup>[47]</sup> To implement locking, SQL Server contains the *Lock Manager*. The Lock Manager maintains an in-memory table that manages the database objects and locks, if any, on them along with other metadata about the lock. Access to any shared object is mediated by the lock manager, which either grants access to the resource or blocks it.

SQL Server also provides the optimistic concurrency control mechanism, which is similar to the [multiversion concurrency control](#) used in other databases. The mechanism allows a new version of a row to be created whenever the row is updated, as opposed to overwriting the row, i.e., a row is additionally identified by the ID of the transaction that created the version of the row. Both the old as well as the new versions of the row are stored and maintained, though the old versions are moved out of the database into a system database identified as `Tempdb`. When a row is in the process of being updated, any other requests are not blocked (unlike locking) but are executed on the older version of the row. If the other request is an update statement, it will result in two different versions of the rows—both of them will be stored by the database, identified by their respective transaction IDs.<sup>[47]</sup>

## Data retrieval and programmability [\[edit\]](#)

The main mode of retrieving data from an SQL Server database is [querying](#) for it. The query is expressed using a variant of [SQL](#) called [T-SQL](#), a dialect Microsoft SQL Server shares with [Sybase SQL Server](#) due to its legacy. The query [declaratively](#) specifies what is to be retrieved. It is processed by the query processor, which figures out the sequence of steps that will be necessary to retrieve the requested data. The sequence of actions necessary to execute a query is called a [query plan](#). There might be multiple ways to process the same query. For example, for a query that contains a [join](#) statement and a [select](#) statement, executing join on both the tables and then executing select on the results would give the same result as selecting from each table and then executing the join, but result in different execution plans. In such case, SQL Server chooses the plan that is expected to yield the results in the shortest possible time. This is called [query optimization](#) and is performed by the query processor itself.<sup>[47]</sup>

SQL Server includes a cost-based query optimizer which tries to optimize on the cost, in terms of the resources it will take to execute the query. Given a query, then the query optimizer looks at the [database schema](#), the database statistics and the system load at that time. It then decides which sequence to access the tables referred in the query, which sequence to execute the operations and what access method to be used to access the tables. For example, if the table has an associated index, whether the index should be used or not: if the index is on a column which is not unique for most of the columns (low "selectivity"), it might not be worthwhile to use the index to access the data. Finally, it decides whether to execute the query [concurrently](#) or not. While a concurrent execution is more costly in terms of total processor time, because the execution is actually split to different processors might mean it will execute faster. Once a query plan is generated for a query, it is temporarily cached. For further invocations of the same query, the cached plan is used. Unused plans are discarded after some time.<sup>[47][63]</sup>

SQL Server also allows [stored procedures](#) to be defined. Stored procedures are parameterized T-SQL queries, that are stored in the server itself (and not issued by the client application as is the case with general queries). Stored procedures can accept values sent by the client as input parameters, and send back results as output parameters. They can call defined functions, and other stored procedures, including the same stored procedure (up to a set number of times). They can be [selectively provided access to](#). Unlike other queries, stored procedures have an associated name, which is used at runtime to resolve into the actual queries. Also because the code need not be sent from the client every time (as it can be accessed by name), it reduces network traffic and somewhat improves performance.<sup>[64]</sup> Execution plans for stored procedures are also cached as necessary.

## T-SQL [\[edit\]](#)

*Main article: T-SQL*

T-SQL (Transact-SQL) is the secondary means of programming and managing SQL Server. It exposes keywords for the operations that can be performed on SQL Server, including creating and altering database schemas, entering and editing data in the database as well as monitoring and managing the server itself. Client applications that consume data or manage the server will leverage SQL Server functionality by sending T-SQL queries and statements which are then processed by the server and results (or errors) returned to the client application. SQL Server allows it to be managed using T-SQL. For this it exposes read-only tables from which server statistics can be read. Management functionality is exposed via system-defined stored procedures which can be invoked from T-SQL queries to perform the management operation. It is also possible to create linked Servers using T-SQL. Linked servers allow a single query to process operations performed on multiple servers.<sup>[65]</sup>

## SQL Native Client (aka SNAC) [ edit ]

SQL Native Client is the native client side data access [library](#) for Microsoft SQL Server, version 2005 onwards. It natively implements support for the SQL Server features including the [Tabular Data Stream](#) implementation, support for mirrored SQL Server databases, full support for all data types supported by SQL Server, asynchronous operations, query notifications, [encryption](#) support, as well as receiving multiple result sets in a single database session. SQL Native Client is used under the hood by SQL Server plug-ins for other data access technologies, including [ADO](#) or [OLE DB](#). The SQL Native Client can also be directly used, bypassing the generic data access layers.<sup>[66]</sup>

On November 28, 2011, a preview release of the SQL Server [ODBC](#) driver for [Linux](#) was released.<sup>[67]</sup>

## SQL CLR [ edit ]

*Main article: [SQL CLR](#)*

Microsoft SQL Server 2005 includes a component named [SQL CLR](#) ("Common Language Runtime") via which it integrates with [.NET Framework](#). Unlike most other applications that use [.NET Framework](#), SQL Server itself hosts the [.NET Framework runtime](#), i.e., memory, threading and resource management requirements of [.NET Framework](#) are satisfied by [SQLOS](#) itself, rather than the underlying Windows operating system. [SQLOS](#) provides deadlock detection and resolution services for [.NET](#) code as well. With [SQL CLR](#), [stored procedures](#) and [triggers](#) can be written in any [managed .NET language](#), including [C#](#) and [VB.NET](#). [Managed code](#) can also be used to define [UDT's \(user defined types\)](#), which can persist in the database. [Managed code](#) is compiled to [CLI assemblies](#) and after being verified for [type safety](#), registered at the database. After that, they can be invoked like any other procedure.<sup>[68]</sup> However, only a subset of the [Base Class Library](#) is available, when running code under [SQL CLR](#). Most APIs relating to [user interface](#) functionality are not available.<sup>[68]</sup>

When writing code for [SQL CLR](#), data stored in [SQL Server](#) databases can be accessed using the [ADO.NET APIs](#) like any other [managed application](#) that accesses [SQL Server](#) data. However, doing that creates a new database session, different from the one in which the code is executing. To avoid this, [SQL Server](#) provides some enhancements to the [ADO.NET](#) provider that allows the connection to be redirected to the same session which already hosts the running code. Such connections are called context connections and are set by setting [context connection](#) parameter to [true](#) in the connection string. [SQL Server](#) also provides several other enhancements to the [ADO.NET](#) API, including classes to work with tabular data or a single row of data as well as classes to work with internal metadata about the data stored in the database. It also provides access to the [XML](#) features in [SQL Server](#), including [XQuery](#) support. These enhancements are also available in [T-SQL](#) Procedures in consequence of the introduction of the new [XML Datatype](#) ([query.value.nodes](#) functions).<sup>[69]</sup>

## Services [ edit ]

[SQL Server](#) also includes an assortment of add-on services. While these are not essential for the operation of the database system, they provide value added services on top of the core database management system. These services either run as a part of some [SQL Server](#) component or out-of-process as [Windows Service](#) and presents their own [API](#) to control and interact with them.

### Service Broker [ edit ]

Used inside an instance, programming environment. For cross instance applications, Service Broker communicates over [TCP/IP](#) and allows the different components to be synchronized together, via exchange of messages. The Service Broker, which runs as a part of the database engine, provides a reliable messaging and [message queuing](#) platform for [SQL Server](#) applications.<sup>[70]</sup>

### Replication Services [ edit ]

[SQL Server](#) Replication Services are used by [SQL Server](#) to replicate and synchronize database objects, either in entirety or a subset of the objects present, across replication agents, which might be other database servers across the network, or database caches on the client side. Replication follows a publisher/subscriber model, i.e., the changes are sent out by one database server ("publisher") and are received by others ("subscribers"). [SQL Server](#) supports three different types of replication.<sup>[71]</sup>

#### Transaction replication

Each transaction made to the publisher database (master database) is synced out to subscribers, who update their databases with the transaction. Transactional replication synchronizes databases in near real time.<sup>[72]</sup>

#### Merge replication

Changes made at both the publisher and subscriber databases are tracked, and periodically the changes are synchronized bi-directionally between the publisher and the subscribers. If the same data has been modified differently in both the publisher and the subscriber databases, synchronization will result in a conflict which has to be resolved, either manually or by using pre-defined policies. [rowguid](#) needs to be configured on a column if merge replication is configured.<sup>[73]</sup>

#### Snapshot replication

Snapshot replication publishes a copy of the entire database (the then-snapshot of the data) and replicates out to the subscribers. Further changes to the snapshot are not tracked.<sup>[74]</sup>

## Analysis Services [ edit ]

*Main article: [SQL Server Analysis Services](#)*

[SQL Server](#) Analysis Services adds [OLAP](#) and [data mining](#) capabilities for [SQL Server](#) databases. The [OLAP](#) engine supports [MOLAP](#), [ROLAP](#) and [HOLAP](#) storage modes for data. Analysis Services supports the [XML for Analysis](#) standard as the underlying communication protocol. The cube data can be accessed using [MDX](#) and [LINQ](#)<sup>[75]</sup> queries.<sup>[76]</sup> Data mining specific functionality is exposed via the [DMX](#) query language. Analysis Services includes various algorithms—[Decision trees](#), clustering algorithm, [Naive Bayes](#) algorithm, time series analysis, sequence clustering algorithm, linear and logistic [regression](#) analysis, and [neural networks](#)—for use in data mining.<sup>[77]</sup>

## Reporting Services [ edit ]

*Main article: [SQL Server Reporting Services](#)*

[SQL Server](#) Reporting Services is a report generation environment for data gathered from [SQL Server](#) databases. It is administered via a [web interface](#). Reporting services features a [web services](#) interface to support the development of custom reporting applications. Reports are created as [RDL](#) files.<sup>[78]</sup>

Reports can be designed using recent versions of [Microsoft Visual Studio](#) (Visual Studio.NET 2003, 2005, and 2008)<sup>[79]</sup> with [Business Intelligence Development Studio](#), installed or with the included [Report Builder](#). Once created, [RDL](#) files can be rendered in a variety of formats,<sup>[80]</sup><sup>[81]</sup> including [Excel](#), [PDF](#), [CSV](#), [XML](#), [BMP](#), [EMF](#), [GIF](#), [JPEG](#), [PNG](#), and [TIFF](#),<sup>[82]</sup> and [HTML](#) Web Archive.

## Notification Services [ edit ]

*Main article: [SQL Server Notification Services](#)*

Originally introduced as a post-release add-on for [SQL Server](#) 2000,<sup>[83]</sup> Notification Services was bundled as part of the Microsoft [SQL Server](#) platform for the first and only time with [SQL Server](#) 2005.<sup>[84]</sup><sup>[85]</sup> [SQL Server](#) Notification Services is a mechanism for generating data-driven notifications, which are sent to Notification Services subscribers. A subscriber registers for a specific event or transaction (which is registered on the database server as a trigger); when the event occurs, Notification Services can use one of three methods to send a message to the subscriber informing about the occurrence of the event. These methods include [SMTP](#), [SOAP](#), or by writing to a file in the filesystem.<sup>[86]</sup> Notification Services was discontinued by Microsoft with the release of [SQL Server](#) 2008 in August 2008, and is no longer an officially supported component of the [SQL Server](#) database platform.

## Integration Services [ edit ]

*Main article: [SQL Server Integration Services](#)*

[SQL Server](#) Integration Services (SSIS) provides [ETL](#) capabilities for [SQL Server](#) for [data import](#), [data integration](#) and [data warehousing](#) needs. Integration Services includes GUI tools to build [workflows](#) such as extracting data from various sources, querying data, transforming data—including aggregation, de-duplication, de-normalization and merging of data—and then exporting the transformed data into destination databases or files.<sup>[87]</sup>

## Full Text Search Service [ edit ]

[SQL Server](#) Full Text Search service is a specialized indexing and querying service for unstructured text stored in [SQL Server](#) databases. The full text search index can be created on any column with character based text data. It allows for words to be searched for in the text columns. While it can be performed with the [SQL LIKE](#) operator, using [SQL Server](#) Full Text Search service can be more efficient. Full allows for inexact matching of the source string, indicated by a *Rank* value which can range from 0 to 1000 - a higher rank means a more accurate match. It also allows linguistic matching ("inflectional search"), i.e., linguistic variants of a word (such as a verb in a different tense) will also be a match for a given word (but with a lower rank than an exact match). Proximity searches are also supported, i.e., if the words searched for do not occur in the sequence they are specified in the query but are near each other, they are also considered a match. [T-SQL](#) exposes special operators that can be used to access the FTS capabilities.<sup>[88]</sup><sup>[89]</sup>

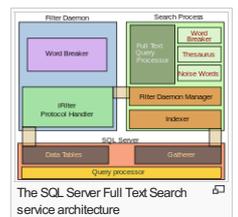
The Full Text Search engine is divided into two processes: the *Filter Daemon* process ([msftfd.exe](#)) and the *Search* process ([msftesql.exe](#)). These processes interact with the [SQL Server](#). The Search process includes the indexer (that creates the full text indexes) and the full text query processor. The indexer scans through text columns in the database. It can also index through binary columns, and use [iFilters](#) to extract meaningful text from the binary blob (for example, when a [Microsoft Word](#) document is stored as an unstructured binary file in a database). The [iFilters](#) are hosted by the Filter Daemon process. Once the text is extracted, the Filter Daemon process breaks it up into a sequence of words and hands it over to the indexer. The indexer filters out *noise words*, i.e., words like *A*, *And* etc., which occur frequently and are not useful for search. With the remaining words, an [inverted index](#) is created, associating each word with the columns they were found in. [SQL Server](#) itself includes a *Gatherer* component that monitors changes to tables and invokes the indexer in case of updates.<sup>[90]</sup>

When a full text query is received by the [SQL Server](#) query processor, it is handed over to the FTS query processor in the Search process. The FTS query processor breaks up the query into the constituent words, filters out the noise words, and uses an inbuilt [thesaurus](#) to find out the linguistic variants for each word. The words are then queried against the inverted index and a rank of their accurateness is computed. The results are returned to the client via the [SQL Server](#) process.<sup>[90]</sup>

## SQLCMD [ edit ]

[SQLCMD](#) is a command line application that comes with Microsoft [SQL Server](#), and exposes the management features of [SQL Server](#). It allows [SQL](#) queries to be written and executed from the command prompt. It can also act as a [scripting language](#) to create and run a set of [SQL](#) statements as a script. Such scripts are stored as a [.sql](#) file, and are used either for management of databases or to create the database schema during the deployment of a database.

[SQLCMD](#) was introduced with [SQL Server](#) 2005 and this continues with [SQL Server](#) 2012 and 2014. Its predecessor for earlier versions was [OSQL](#) and [ISQL](#), which is functionally equivalent as it pertains to [TSQL](#) execution, and many of the command line parameters are identical, although [SQLCMD](#) adds extra versatility.



The [SQL Server](#) Full Text Search service architecture

## Visual Studio [ edit ]

Main article: *Microsoft Visual Studio*

**Microsoft Visual Studio** includes native support for data programming with Microsoft SQL Server. It can be used to write and **debug code** to be executed by SQL CLR. It also includes a *data designer* that can be used to graphically create, view or edit database schemas. Queries can be created either visually or using code. SSMS 2008 onwards, provides **intellisense** for SQL queries as well.

## SQL Server Management Studio [ edit ]

Main article: *SQL Server Management Studio*

**SQL Server Management Studio** is a GUI tool included with SQL Server 2005 and later for configuring, managing, and administering all components within Microsoft SQL Server. The tool includes both script editors and graphical tools that work with objects and features of the server.<sup>[91]</sup> SQL Server Management Studio replaces **Enterprise Manager** as the primary management interface for Microsoft SQL Server since SQL Server 2005. A version of SQL Server Management Studio is also available for SQL Server Express Edition, for which it is known as *SQL Server Management Studio Express* (SSMSE).<sup>[92]</sup>

A central feature of SQL Server Management Studio is the Object Explorer, which allows the user to browse, select, and act upon any of the objects within the server.<sup>[93]</sup> It can be used to visually observe and analyze query plans and optimize the database performance, among others.<sup>[94]</sup> SQL Server Management Studio can also be used to create a new database, alter any existing database schema by adding or modifying tables and indexes, or analyze performance. It includes the query windows which provide a GUI based interface to write and execute queries.<sup>[47]</sup>

## Business Intelligence Development Studio [ edit ]

Main article: *Business Intelligence Development Studio*

**Business Intelligence Development Studio** (BIDS) is the **IDE** from **Microsoft** used for developing data analysis and **Business Intelligence** solutions utilizing the Microsoft **SQL Server Analysis Services**, **Reporting Services** and **Integration Services**. It is based on the **Microsoft Visual Studio** development environment but is customized with the SQL Server services-specific extensions and project types, including tools, controls and projects for reports (using Reporting Services), Cubes and **data mining** structures (using Analysis Services).<sup>[95]</sup> For SQL Server 2012 and later, this IDE has been renamed SQL Server Data Tools (SSDT).

## See also [ edit ]

- Comparison of relational database management systems
- Comparison of object-relational database management systems
- Comparison of data modeling tools
- List of relational database management systems

## References [ edit ]

- ↑ "SQL Server 2014 released to manufacturers, will be generally available April 1". *Microsoft Official Blog*. Microsoft. March 18, 2014. Retrieved June 9, 2014.
- ↑ Lextrait, Vincent (July 2010). "The Programming Languages Beacon, v10.3". Retrieved September 5, 2010.
- ↑ "Download Microsoft SQL Server 2008 R2". *Microsoft Evaluation Center*. Microsoft Corporation. Retrieved July 18, 2011.
- ↑ Harris, Scott; Curtis Preston (2007). *Backup & Recovery: Inexpensive Backup Solutions for Open Systems*. O'Reilly, p. 562. ISBN 0596102461.
- ↑ All about the History of SQL Server - Scriptcase Blog PHP. Scriptcase.net. Retrieved on 2014-03-23
- ↑ "Database Engine XML Enhancements". Retrieved 2007-12-03.
- ↑ "Database Engine Enhancements". Retrieved 2007-12-03.
- ↑ Multiple Active Result Sets (MARS) in SQL Server 2005. retrieved June 20, 2009
- ↑ Dynamic Management Views and Functions. retrieved June 6, 2010
- ↑ "Issues to consider when you use the database mirroring feature in the initial release of SQL Server 2005". Support.microsoft.com. 2007-11-20. Retrieved 2011-09-04.
- ↑ "Automatic Failover". Msdn.microsoft.com. Retrieved 2011-09-04.
- ↑ "Microsoft SQL Server 2008". Retrieved 2007-04-06.
- ↑ "ChannelWeb: Next SQL Server stop: Katmai". Retrieved 2005-11-05.
- ↑ "Microsoft Gives Peek At Next Version Of SQL Server". Retrieved 2007-05-11.
- ↑ "One more test build to go for SQL Server 2008". Retrieved 2006-11-13.
- ↑ "Guest Blogger: Ted Kummert". Retrieved 2007-11-20.
- ↑ Christian Kleinerman. "SQL Server 2008 for developers". Channel 9. Retrieved 2008-03-07.
- ↑ Fernando Azpeitia Lopez. "SQL Server 2008 Full-Text Search: Internals and Enhancements".
- ↑ Microsoft Shares Details on SQL Server 2008 Spatial Support by Directions Staff. Archived from the original on 2007-08-08. Retrieved 2007-09-07.
- ↑ "Features of SQL 2008". Technet.microsoft.com. 2009-02-12. Retrieved 2011-09-04.
- ↑ "SQL Server "Katmai" to Deliver Entity Data Platform and Support LINQ". Retrieved 2007-05-12.
- ↑ Microsoft Details Dynamic IT Strategy at Tech-Ed 2007. Retrieved 2007-06-04.
- ↑ "SQL Server IntelliSense". Retrieved 2008-08-18.
- ↑ "SQL Server Support for PowerShell". Retrieved 2007-12-03.
- ↑ "SQL Server 2008 Service Pack 4 has released". *SQL Release Services Blog*. Retrieved 2014-10-01.
- ↑ "SQL Server 2008 R2 Launches". Retrieved 2010-04-21.
- ↑ SQL Server Team. "TechEd 2009 – new SQL Server Innovations". MSDN Blogs. Retrieved 2009-05-12.
- ↑ "Review: Microsoft SQL Server 2008 R2". Networkworld.com. 2010-05-03. Retrieved 2013-06-15.
- ↑ "SQL Server 2008 R2 Application and Multi-Server Management". Retrieved 2010-06-06.
- ↑ "Microsoft® SQL Server® 2008 R2 SP1". Retrieved 2011-09-19.
- ↑ "Microsoft® SQL Server® 2008 R2 SP2". Retrieved 2012-07-31.
- ↑ "SQL Server 2008 R2 Service Pack 3 has released". *SQL Release Services Blog*. Retrieved 2014-10-01.
- ↑ "Microsoft Releases SQL Server 2012 to Help Customers Manage "Any Data, Any Size, Anywhere"". *Microsoft News Center*. Microsoft. March 6, 2012. Retrieved March 7, 2012.
- ↑ Lam, Rohan. "Microsoft is Aligning with ODBC for Native Relational Data Access — FAQ". *SQL Server Forums*. Microsoft Corporation. Retrieved March 7, 2012.
- ↑ "Availability Enhancements (Database Engine)". Msdn.microsoft.com. Retrieved 2013-06-15.
- ↑ "Manageability Enhancements (Database Engine)". Msdn.microsoft.com. 2011-07-13. Retrieved 2013-06-15.
- ↑ Alastair Aitchison (2012). *Pro Spatial with SQL Server 2012*. Apress. pp. 21–23. ISBN 978-1-4302-3491-3.
- ↑ "Programmability Enhancements (Database Engine)". Msdn.microsoft.com. Retrieved 2013-06-15.
- ↑ "Scalability and Performance Enhancements (Database Engine)". Msdn.microsoft.com. Retrieved 2013-06-15.
- ↑ "Security Enhancements (Database Engine)". Msdn.microsoft.com. Retrieved 2013-06-15.
- ↑ SQL Server 2014 - CTP2 is now available. MSDN Blogs. Retrieved on 2014-03-23
- ↑ SQL Server 2014-Explore. Server Cloud. Microsoft.com (2013-04-17). Retrieved on 2014-03-23
- ↑ Laden Sie die Testversion herunter: Microsoft SQL Server 2014 (in German). Microsoft. Retrieved 2014-11-05.
- ↑ "SQL Server 2014 Service Pack 1 release information". *Microsoft.com*. Microsoft, Inc. Retrieved 17 May 2015.
- ↑ "SQL Server 2016 Release Notes". *msdn.microsoft.com*. Retrieved 2016-01-04.
- ↑ "Compare Editions". *SQL Server homepage*. Microsoft Corporation. Retrieved 2007-12-03.
- ↑ *abcde fgh i* Kalen Delaney. *Inside Microsoft SQL Server 2005: The Storage Engine*. Microsoft Press. ISBN 0-7356-2105-5.
- ↑ "Choosing a StreamInsight Edition". *MSDN*. Microsoft Corporation. Retrieved July 18, 2011.
- ↑ <sup>[1]</sup>
- ↑ "SQL Server 2008: Editions". Retrieved 2011-07-21.
- ↑ "Database System | Performance & Scalability | SQL Server 2012 Business Intelligence Editions". Microsoft.com. Retrieved 2013-06-15.
- ↑ "SQL Server 2008 R2 Express Database Size Limit Increased to 10GB". Retrieved 2010-04-23.
- ↑ "What's up with SQL Server 2008 Express editions". Retrieved 2008-08-15.
- ↑ "Developer Edition". *SQL Server home*. Microsoft Corporation. Retrieved July 18, 2011.
- ↑ "SQL Server 2008 Trial Software". Retrieved 2009-03-26.
- ↑ "Microsoft SQL Server 2008: Fast Track Data Warehouse". Retrieved 2009-03-26.
- ↑ "SQL Server 2012 Express LocalDB". msdn.microsoft.com. Retrieved March 11, 2013.
- ↑ "Introducing LocalDB, an improved SQL Express".
- ↑ "Microsoft Analytics Platform System". Retrieved 2015-04-29.
- ↑ "Pages and Extents". Retrieved 2007-12-02.
- ↑ "Table and Index Organization". Retrieved 2007-12-02.
- ↑ "Buffer Management". Retrieved 2007-12-02.
- ↑ "Single SQL Statement Processing". Retrieved 2007-12-03.
- ↑ "Stored Procedure Basics". Retrieved 2007-12-03.
- ↑ "Transact-SQL Reference". Retrieved 2007-12-03.
- ↑ "Features of SQL Native Client". Retrieved 2007-12-03.
- ↑ "Available Today: Preview Release of the SQL Server ODBC Driver for Linux". SQL Server Team Blog. 2011-11-28. Retrieved 2013-06-15.
- ↑ "Overview of CLR integration". Retrieved 2007-12-03.
- ↑ "XML Support in SQL Server". Retrieved 2008-09-05.
- ↑ "Introducing Service Broker". Retrieved 2007-12-03.
- ↑ "Types of Replication Overview". Retrieved 2007-12-03.
- ↑ "Transactional Replication Overview". Retrieved 2007-12-03.

73. <sup>^</sup> ["Merge Replication Overview"](#)<sup>ⓘ</sup>. Retrieved 2007-12-03.
74. <sup>^</sup> ["Snapshot replication Overview"](#)<sup>ⓘ</sup>. Retrieved 2007-12-03.
75. <sup>^</sup> ["SSAS Entity Framework Provider"](#)<sup>ⓘ</sup>. Retrieved 2011-09-29.
76. <sup>^</sup> ["Analysis Services Architecture"](#)<sup>ⓘ</sup>. Retrieved 2007-12-03.
77. <sup>^</sup> ["Data Mining Concepts"](#)<sup>ⓘ</sup>. Retrieved 2007-12-03.
78. <sup>^</sup> ["SQL Server Reporting Services"](#)<sup>ⓘ</sup>. Retrieved 2007-12-03.
79. <sup>^</sup> ["Cannot open a SQL Reporting Services .rptproj file | Microsoft Connect"](#)<sup>ⓘ</sup>. Connect.microsoft.com. Retrieved 2011-09-04.
80. <sup>^</sup> [MSDN Library: Reporting Services Render Method](#)<sup>ⓘ</sup>
81. <sup>^</sup> [Device Information Settings](#)<sup>ⓘ</sup>
82. <sup>^</sup> [Image Device Information Settings](#)<sup>ⓘ</sup>
83. <sup>^</sup> ["An Introduction to SQL Server Notification Services"](#)<sup>ⓘ</sup>. Retrieved 2008-11-14.
84. <sup>^</sup> ["SQL Server Notification Services Removed from SQL Server 2008"](#)<sup>ⓘ</sup>. Retrieved 2008-09-17.
85. <sup>^</sup> ["Discontinued Functionality in SQL Server 2008 Reporting Services"](#)<sup>ⓘ</sup>. Retrieved 2008-09-17.
86. <sup>^</sup> ["Introducing SQL Server Notification Services"](#)<sup>ⓘ</sup>. Retrieved 2007-12-03.
87. <sup>^</sup> ["Integration Services Overview"](#)<sup>ⓘ</sup>. Retrieved 2007-12-03.
88. <sup>^</sup> ["Introduction to Full-Text Search"](#)<sup>ⓘ</sup>. Retrieved 2007-12-03.
89. <sup>^</sup> ["Querying SQL Server using Full-Text Search"](#)<sup>ⓘ</sup>. Retrieved 2007-12-03.
90. <sup>^</sup> <sup>^</sup> ["Full-Text Search Architecture"](#)<sup>ⓘ</sup>. Retrieved 2007-12-03.

## Further reading [[edit](#)]

- Lance Delano, Rajesh George et al. (2005). *Wrox's SQL Server 2005 Express Edition Starter Kit (Programmer to Programmer)*. Microsoft Press. ISBN 0-7645-8923-7
- Delaney, Kalen, et al. (2007). *Inside SQL Server 2005: Query Tuning and Optimization*. Microsoft Press. ISBN 0-7356-2196-9.
- Ben-Gan, Itzik, et al. (2006). *Inside Microsoft SQL Server 2005: T-SQL Programming*. Microsoft Press. ISBN 0-7356-2197-7.

## External links [[edit](#)]

- [Official website](#)<sup>ⓘ</sup>
- [2nd official website at Microsoft TechNet](#)<sup>ⓘ</sup>

 Wikimedia Commons has media related to *Microsoft SQL Server*.

 Wikibooks has a book on the topic of *Microsoft SQL Server*

v · t · e		Database management systems
<b>Types</b>		Object-oriented (comparison) · Relational (comparison) · Document-oriented · Graph · NoSQL · NewSQL
<b>Concepts</b>		Database · ACID · CAP theorem · CRUD · Null · Candidate key · Foreign key · Primary key · Superkey · Surrogate key · Armstrong's axioms
<b>Objects</b>		Relation (table · column · row) · View · Transaction · Log · Trigger · Index · Stored procedure · Cursor · Partition
<b>Components</b>		Concurrency control · Data dictionary · JDBC · XQL · ODBC · Query language · Query optimizer · Query plan
<b>Functions</b>		Administration and automation · Query optimization · Replication
<b>Related topics</b>		Database models · Database normalization · Database storage · Distributed DBMS · Federated database system · Referential integrity · Relational algebra · Relational calculus · Relational database · Relational DBMS · Relational model · Object-relational database · Transaction processing
v · t · e		Microsoft development tools
<b>Development environments</b>	<b>Visual Studio</b> · Visual Studio Code · Express · Visual Studio Lab Management · Visual Studio Test Professional · Tools for Applications · Tools for Office · VSTS Profiler	
	<b>Others</b> · App Studio · Blend · Expression Web · GW-BASIC · MSBuild · Robotics Developer Studio · SharePoint Designer (FrontPage) · PowerShell ISE · QuickBASIC (QBasic) · Small Basic · WebMatrix · Windows SDK	
<b>Languages</b>	Microsoft Basic · Visual Basic · Visual Basic .NET · Visual C++ (C++/CLI · Managed C++) · C# (Visual C#) · F# · J# · Visual FoxPro · IronPython · IronRuby · JavaScript · PowerShell · TypeScript · XML · VBScript	
<b>Major APIs and frameworks (See all)</b>	Windows API · Silverlight · XNA · DirectX (Managed DirectX) · Xbox Development Kit · Windows CardSpace · Windows Installer	
	.NET Framework · ASP.NET · ASP.NET AJAX · ADO.NET · Communication Foundation · Identity Foundation · LINQ · Presentation Foundation · Workflow Foundation	
	Windows Driver Frameworks · Windows Driver Model · KMDF · UMDF · Windows Driver Kit · Windows Hardware Certification Kit	
<b>Database</b>	<b>SQL Server</b> · Express · Compact · Management Studio · Business Intelligence Development Studio · MSDE SQL services · Analysis · Reporting · Integration · Notification Other · Visual FoxPro · Microsoft Access · Jet Engine · Extensible Storage Engine · Azure DocumentDB	
<b>Source control</b>	Team Foundation Server · Visual SourceSafe	
<b>Delivery</b>	End-user · ClickOnce · Windows Installer (WIX) · Windows Store Developers · NuGet · Web Platform	
<b>Performance Analysis</b>	CLR Profiler	
v · t · e		Microsoft
History · All topics		
<b>People</b> (list)	<b>Founders</b> · Bill Gates · Paul Allen <b>Board of directors</b> · John W. Thompson (Chairman) · Satya Nadella (CEO) · Dina Dublon · Maria Klawe · David Marquardt · Charles Noski · Helmut Panke · Mason Merritt · John W. Stanton <b>Senior Leadership Team</b> · Satya Nadella (CEO) · Scott Guthrie · Amy Hood (CFO) · Qi Lu · Terry Myerson · Harry Shum · Kevin Turner (COO) <b>Corporate VPs</b> · Joe Belfiore · Richard Rashid (SVP) · S. Somasegar (SVP)	
<b>Divisions</b>	Engineering groups (Mobile) · Digital Crimes Unit · Garage · Press · Research · Studios · .NET Foundation · Outercurve Foundation	
<b>Regional branches</b>	Microsoft Algeria · Egypt · India · Japan · Pakistan	
<b>Product families</b>	Operating systems (Windows) · Software (Office · Servers · Visual Studio) · Hardware (Band · HoloLens · Lumia · Surface · Xbox)	
<b>Web properties</b>	Bing · Channel 9 · CodePlex · Developer Network · MSN · Office.com · OneDrive · Outlook.com · TechNet · Windows Store	
<b>Conferences</b>	Build · MIX · PDC · TechEd · WinHEC · WPC	
<b>Criticism</b>	Bundling of Microsoft Windows · iLoo · Internet Explorer · _NSAKEY · Windows (2000 · ME · XP · Vista) · Xbox360	
<b>Litigation</b>	<i>Alcatel-Lucent v. Microsoft</i> · <i>Apple v. Microsoft</i> · European Union Microsoft competition case · <i>Microsoft v. Lindows</i> · <i>Microsoft vs. MikeRoweSoft</i> · <i>Microsoft v. Shah</i> · <i>United States v. Microsoft</i>	
<b>Acquisitions</b>	6Wunderkinder · Altamira Software · aQuantive · Azxol · The Blue Ribbon SoundWorks · Bungie · Calista Technologies · Colloquis · Connectix · Consumers Software · Danger · Farecast · FASA Studio · Fast Search & Transfer · Firefly · Forethought · GIANT Company Software · GreenButton · Groove Networks · Hotmail · Jellyfish.com · LinkExchange · Lionhead Studios · Massive Incorporated · Mobile Data Labs · Mojang · Nokia Devices and Services · Onfolio · Pando Networks · PlaceWare · Powerset · ProClarity · Rare · Revolution Analytics · ScreenTonic · Secure Islands · Skype · Sunrise Atelier · Winternals Software · Telego · Telekinetics Research · Tellme Networks · Twisted Pixel Games · Vermeer Technologies · Vizio Corporation · Vivaty (defunct) · VolvoMetric · VxTreme · WebTV Networks · Yammer · Yupi	
<span> Category</span> · <span> Commons</span> · <span> Microsoft portal</span> · <span> WikiProject</span>		

Categories: [Database management systems](#) | [Windows Server System](#) | [Microsoft database software](#) | [Relational database management systems](#)

This page was last modified on 8 January 2016, at 21:48.

Text is available under the [Creative Commons Attribution-ShareAlike license](#); additional terms may apply. By using this site, you agree to the [Terms of Use](#) and [Privacy Policy](#). Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.

[Privacy policy](#) · [About Wikipedia](#) · [Disclaimers](#) · [Contact Wikipedia](#) · [Mobile view](#) · [Developers](#) · [Cookie statement](#)

