

.NET for SQL Server Experts

Now that a lot of the .NET dust has settled, IT professionals are faced with the challenge of making it real within their organisations, as like it or not we will all be going the .NET route at sometime in the future. Unless of course you are a LINUX/UNIX fan but I imagine not many of them read ESM each month.

.NET has huge implications for developers as they are faced with updating their skills to utilise the .NET Framework, learn new languages such as VB.NET or C# and get to grips with ASP.NET.

SQL Server experts have been somewhat ignored in the recent .NET flurry as, quite simply, there isn't a lot to be said about SQL Server at the moment. Our time will come in early 2004 when I expect to see YUKON, the next major version of SQL Server, finally available. Until then we just need to focus on our day jobs of building, managing and enhancing our databases.

That said, SQL Server does have a major part to play today in the world of .NET. It has a significant role in any .NET implementation as almost all of the .NET Enterprise Servers need to store data somewhere, and where better than Microsoft SQL Server.

On that basis I have decided to spend some time taking a look at the key .NET Enterprise Server products and exploring the important role that SQL Server takes when using these for a solution. Everything I mention will be current technologies that I see rolled out across many organisations today.

This month we will start with BizTalk Server and Commerce Server.

BizTalk Server 2002

Microsoft BizTalk Server is designed to provide a central messaging hub which, combined with business process management, makes systems integration a little bit easier. It uses a message based architecture to move business documents and files from one place to another, more often than not spanning quite different technologies and solutions.

A typical example may be an organisation submitting a simple XML (extensible markup language) file by Email to BizTalk Server which in turn will convert it to an SAP iDOC file for submission to an ERP system.

BizTalk Server uses Microsoft SQL Server as its file repository and to store messages that are being stacked for later submission –maybe as the network has temporarily failed.

SQL Server Configuration Tips when used with BizTalk Server

It is quite possible to run BizTalk Server with SQL Server version 7.0 but you must ensure that service pack 3 has been added to the server. SQL Server 2000 requires service pack 1.

Once installed BizTalk Server creates four databases:

- BizTalk Messaging Management database
- Shared Queue database
- Tracking database
- Orchestration Persistence database

To reduce the amount of space required for the databases **Truncate log on checkpoint** and **Auto shrink** should be enabled as the SQL Server logs can grow quite large.

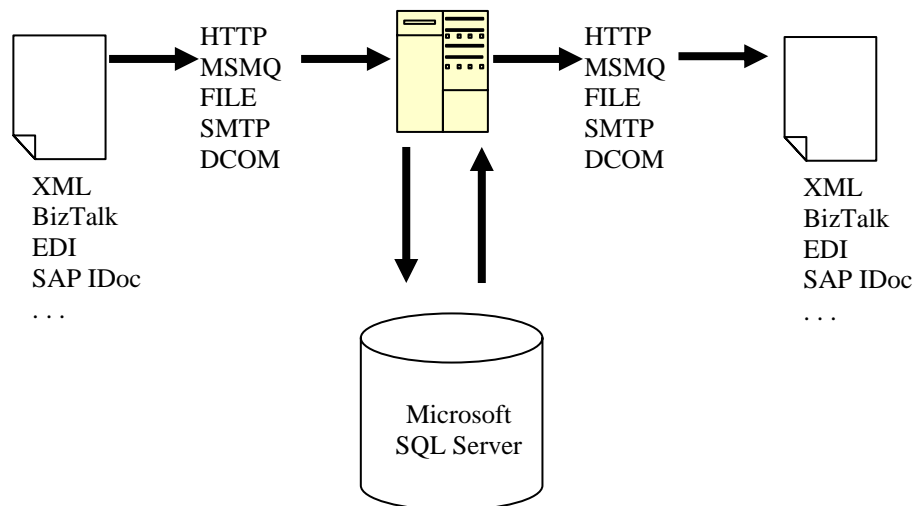


Fig. 1 SQL Server lays at the heart of BizTalk Server

BizTalk will use Windows Authentication to access the shared queue, messaging and tracking databases and the tool in BizTalk used to monitor documents called the Document Tracking Web application.

Once installed the SQL Server login account needs to be configured with permissions to access the BizTalk Messaging Management, Shared Queue, and Tracking databases.

In a cluster environment with SQL Server and BizTalk Server installed on different computers, SQL Server client tools must be installed on the computer where BizTalk Server 2002 is installed.

When you install SQL Server client utilities, SQL Server Setup uses Named Pipes as the default client network library but BizTalk Server uses TCP/IP for communication with SQL Server. The default network library will need to be changed to TCP/IP.

For larger scale applications BizTalk Server is often sited on a different physical server to SQL Server. This allows both to be optimised for the specific hardware, making best use of memory and processors. In this configuration the account you use for both servers must have administrator privileges.

In addition BizTalk Server cannot access SQL Server databases running a different language to itself.

Optimisation of SQL Server in a BizTalk Environment

There really are no surprises here. Some of the areas especially affected include:

- Memory – if the available bytes are less than 10MB then any memory spike will start the server disk paging. Page Faults/sec, Pages Input/sec, and Page Reads/sec should all be low, if not increase the amount of RAM on the server.
- Physical Disk - read/sec and writes/sec should be within the capacity of the disk drive being used. % disk time should be less than 80%. It is possible for this to reach the 1000% level plus, when things are really bad. When over 100% the application is essentially disk bound and new disks need to be added. Likewise % idle time should be greater than 20%. Changes to these parameters are normally made via the RAID system that you are running.
- Current Disk Queue Length - the number of requests outstanding on the disk at the time the performance data is collected. Ideally this counter should average less than 2 for optimum performance.

Microsoft Commerce Server

Commerce Server uses SQL Server to host the various product/sales databases and provide the tools to analyse sales and click stream related data. In fact there are a few hundred SQL Server tables behind each Commerce Server SQL Server setup, some of which are used more frequently than others.

A typical Commerce Server 2002 installation will have a SQL Server database structure similar to this:

- Commerce Server <site name>_commerce. This database contains data used for catalogues, campaigns, transactions, events and profiles.
- Administration MSCS_Admin. This database contains configuration data used by Commerce Server spread across 7 tables.
- Data Warehouse <site name>_dw. Database used for collecting operational data about users, transaction data, and click stream data. This data needs to be imported from Web log files and various Commerce Server databases using SQL Server Data Transformation Services tasks customised for the purpose.
- Direct Mailer. Database for mailing lists, messages, and events.

The MSCS_CatalogScratch database is used by the catalog in Commerce Server to store any persistent or empty tables created by the Catalog API. This database must sit on the production catalog database server and users of the catalog need to have db_owner permissions on it. Product catalog stored procedures will build temporary tables in this database on an as-required basis, with the benefit that empty tables are maintained to ensure reuse of the query plan optimising performance.

A number of these databases contain core tables that must not be modified by developers/DBA's in anyway or you may find that Commerce Server will become corrupted or at the very least behave oddly. You have been warned!

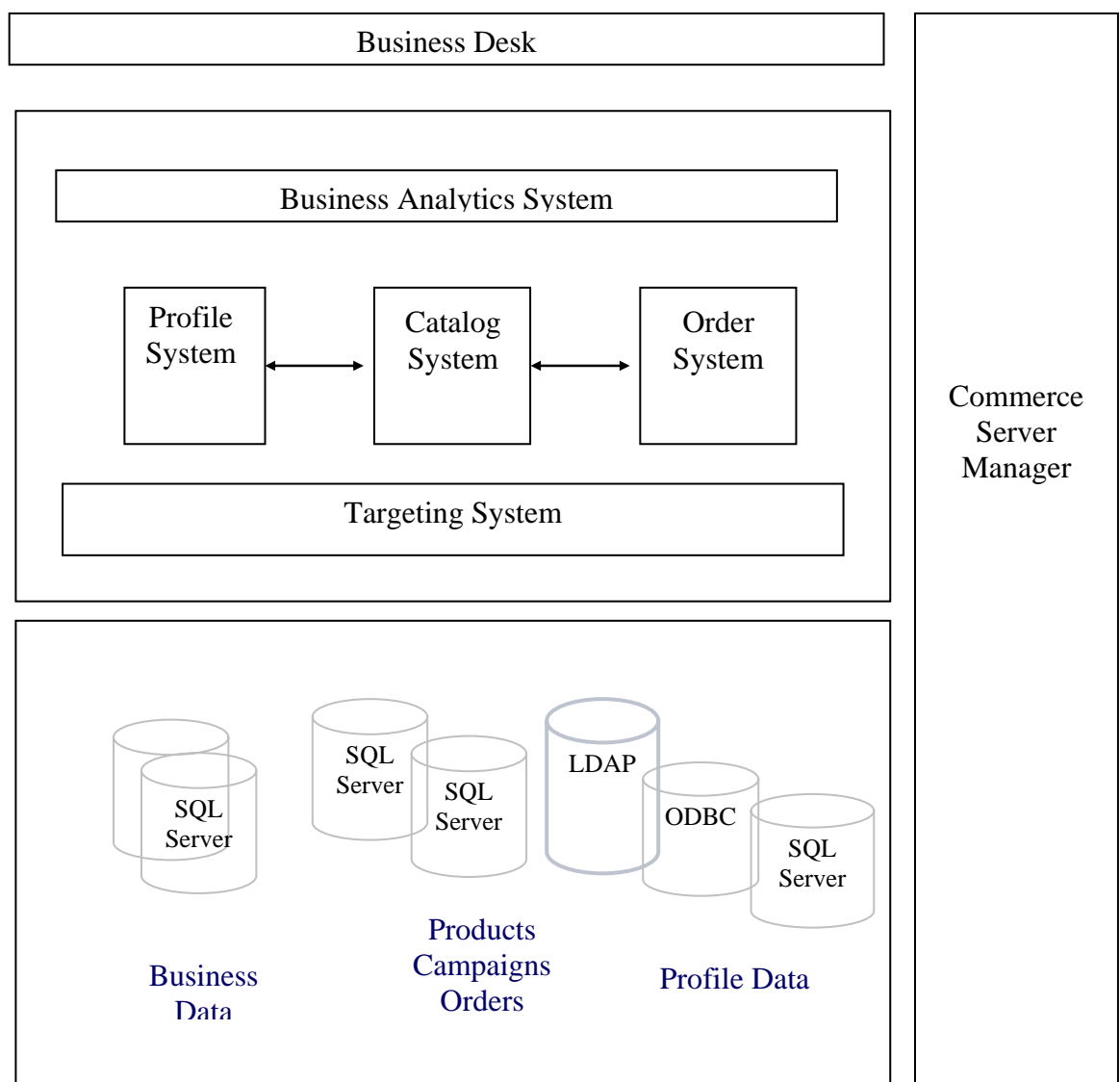


Fig. 2 Commerce Server relies heavily on Microsoft SQL Server

For the most effective SQL Server performance it is now standard practice to separate the business logic tier from the database tier. This applies equally

well to Commerce Server, where the SQL Server used for data storage is separate from the Commerce Server applications and administrative tools. That said, it is quite acceptable to install the Commerce Server Direct Mailer engine on the database server directly.

Clustering is a way of improving system up time. If a server should fail then the work will “fail over” to another server and users continue with their work. This is recommended for a Commerce Server installation, especially if you are providing a 24/7 retail site. The only caveat is that the Commerce Server 2002 Data Warehouse does not support OLAP clustering.

Log shipping is also invaluable as it permits the transfer of a database log from one server to another non-production machine that will improve system availability upon failure of the first system. Remember that this is only a warm standby and there is no guarantee that all transactions will be correctly failed over – missing data is a possibility, especially on sites with high transactional through put.

As Commerce Server so resolutely depends upon SQL Server the associated back up regime of a Commerce Server is the same as for many SQL Server sites, including:

- Keep the transaction log small by backing it up frequently and allowing it to be truncated automatically.
- Always backup the transaction log file before completing any other backups.
- Ensure the business understands the ramification of the back up strategy you are implementing. If they REALLY want 99.999% up time, can they afford it?
- Don't back up tempdb
- Rebuild full text indexes when the associated database is restored
- Ensure you have enough disk space for the backup in question

All database access in Commerce Server should be done using Windows Authentication and all Business Desk and run time users must have appropriate access assigned to the databases.

As you can see from this brief snapshot SQL Server is very tightly integrated into Microsoft .NET. This will undoubtedly increase further with the advent of YUKON, hopefully ensuring that we all have a long and healthy future as SQL Server database experts.