

# SQL SERVER<sup>®</sup> m a g a z i n e

April 2008

## Visual Studio 2008 Unites DATA DEVELOPMENT

p. 12

**LINQ** to Your SQL Server Data p. 17

**DEBUG** Your SSIS Packages p. 21

Dissecting SQL Server's  
**TOP WAITS** p. 25

SET OPERATIONS  
SQL Server 2005  
**SECRETS** Revealed p. 27

T-SQL 101, LESSON 2  
**INSERT, UPDATE, & DELETE** p. 31

**DRILL DEEPER** into  
Plan-Cache Metadata p. 34

SHARPEN YOUR SQL SKILLS  
Surface Area  
Configuration Tools p. 37

LOOK OUT FOR SOME APRIL FOOLERY—CONTEST DETAILS INSIDE!

**PAY UP TO  
72% LESS  
FOR YOUR  
BUSINESS  
INTELLIGENCE**

A series of thin, white, wavy lines that originate from the bottom left and curve upwards and to the right, creating a sense of motion and depth against the black background.

Dell™, Microsoft®, and AMD can help reduce the cost and complexity of Business Intelligence. Our systems are quick to install, easy to manage, and built with standard components that work with what you already use — all for up to 72% less per terabyte than the competition.\*

SIMPLIFY YOUR IT AT [DELL.COM/SQLBI](http://DELL.COM/SQLBI)



\*72% claim based upon a comparison of list prices of typical Business Intelligence offerings from leading hardware manufacturers versus Dell/Microsoft combined offerings. Benchmarked systems configured with 4-5 TB of data storage, database application software, and Business Intelligence analytic software. Dell is a trademark of Dell Inc. ©2008 Dell Inc. All rights reserved.



# 21

Number of silos in which your customer data resides.  
Coincidentally, also the number of totally different  
definitions you have for "customer."

# one

definition —

Embarcadero solutions help manage data enterprise-wide, enabling you to build and communicate an architecture of your data assets so you always know what is what, and where it is. With just one, everything gets simpler. You and your business are empowered to use and reuse reliable, relevant, and timely data across a range of platforms and storage formats. Understanding your data is finally possible, and getting a hold of the metadata is the key. Standards can be defined and enforced. Data redundancy is eliminated. It's no coincidence that the most sophisticated solution is also the simplest. See how one gets it done.

Call 415.834.3131 or visit [www.embarcadero.com](http://www.embarcadero.com)





## COVER STORY

The Next Generation:

# Visual Studio 2008

12

—Michael Otey

Take a tour of the new features and functionality in Visual Studio 2008.



MORE ON THE WEB

### Visual Studio 2008 Bridges Dev and IT

—Karen Forster

Microsoft Senior VP S. Somasegar candidly discusses data-enabled features of Visual Studio, including LINQ and the Entity Data Model, and how Visual Studio supports SQL Server as a data platform.

InstantDoc ID 98342

## FEATURES

17

### LINQ to Your SQL Server Data

—William Sheldon

Build efficient queries using LINQ, which is included in Visual Studio 2008.

21

### Debugging and Logging in SSIS

—Erin Welker

Easily debug packages, Control Flow tasks, and Data Flow tasks as well as log package-execution information using the capabilities in SSIS.

25

### Dissecting SQL Server's Top Waits

—Andrew J. Kelly

It's time to build on your knowledge of wait-stat collection and reporting. Let's focus on a few wait stats that typically account for most overall system waits.

27

### Set Operations

—Itzik Ben-Gan

Circumvent unsupported logical phases, work around SQL Server 2005's lack of INTERSECT ALL and EXCEPT ALL, and take advantage of the INTERSECT and EXCEPT operations' treatment of NULLs.

31

### T-SQL101, Lesson 2

—Bill McEvoy

Learn how to create T-SQL queries that insert new data, update existing data, and delete old data by using a database named MyDB, a table named MyTable, and some dummy tables.

34

### Further Explorations in Plan-Cache Metadata

—Kalen Delaney

Uncover more valuable metadata that can aid you in troubleshooting and performance-tuning your queries.

37

### Sharpen Your Basic SQL Server Skills

—Pinalkumar Dave

Get answers to your questions about SQL Server 2005 Surface Area Configuration tools, and learn how to lower the risk of a security breach.

39

### Data Warehousing: Horizontally Partitioning the Fact Table

—Michelle A. Poolet

Save yourself from having to deal with large, difficult-to-manage fact tables by partitioning the rows.

GAIN CONTROL '08 TOUR

# STRESS FREE CHANGE

ROLL OUT. ROLL BACK. TRACK. ROCK ON.

**QUEST SOFTWARE IS PROUD TO PRESENT CHANGE DIRECTOR FOR SQL SERVER**

FEATURING:

★ **IN-DEPTH CHANGE TRACKING** ★ **AUTOMATED & MANUAL ROLL OUT** ★ **FLEXIBLE ROLL BACK** ★

Take the stage with Quest Software and make your database world rock. Quest gives you the control you need to tune up performance, keep change management pitch perfect and make administration as easy as one-two-three-four.

With Quest, your SQL Server environment *never* misses a beat.

Learn how to automate change management. Read our new technical brief:

"Bullet-proof Change Management - Don't Just Cross Your Fingers and Hope" at [www.quest.com/RockChanges](http://www.quest.com/RockChanges)

**BE A SQL SERVER ROCK STAR WITH QUEST.**



# Contents

WWW.SQLMAG.COM  
APRIL 2008 Vol. 10 • No. 4

## Get FOOLISH— and Win!

Things looking a little silly in this month's issue? Don't worry; it's not you—it's us! In celebration of April Fool's day, we've included some foolery throughout this issue. And if you spot what's different, you could win a one-year VIP Subscription, including exclusive online access to every article ever printed in *Windows IT Pro*, *SQL Server Magazine*, *Exchange* and *Outlook Pro* VIP, *Scripting Pro* VIP, and *Security Pro* VIP; a one-year subscription to your choice of either *Windows IT Pro* or *SQL Server Magazine*; and a VIP CD—updated and delivered twice a year. (If you're already a subscriber to one of our magazines or to only the Pro VIPs, we'll upgrade your existing subscription to include all of our resources.)

Just email a list of what you've found to me at [Christan.Humphries@penton.com](mailto:Christan.Humphries@penton.com) by **April 30th**. Out of the entries with the most correct answers, I'll randomly choose 3 winners. Please include your name, postal address, email address, and phone number in your email message.



—Christan Humphries,  
Your Savvy Assistant

## PRODUCTS

- 43 Comparative Review:  
SQL Server Backup  
Compression Shootout**  
—Allan Hirt  
Learn which SQL Server backup compression tool is right for your organization from this SQL Server expert's detailed comparison.

- 46 New Products**  
Check out new and improved SQL Server—related products from Datanamic Solutions B.V., CNS International, Caber Computing, and Microsoft.

## IN EVERY ISSUE

- 7 Editorial:  
MySQL: Sun's Billion  
Dollar Baby**  
—Michael Otey
- 8 SQLMAG.COM  
Community Dialog:  
Readers Talk Back to  
Community Dialog**  
—Anne Grubb  
Read what your peers think about recent Community Dialogs, and if you're in a gambling mood, check out Itzik Ben-Gan's special "lottery" query.

- 11 Reader to Reader**

- 47 MarketPlace**

- 48 The Back Page:  
SQL Server 2008  
Webcasts**  
—Michael Otey

- 48 Your Savvy Assistant**  
—Christan Humphries

# SQL SERVER magazine

The Hitchhiker's Guide to Building Cosmic Applications

Senior Vice President,  
Technology Media Group  
Group Publisher

Kim Paulsen  
[kpaulsen@windowsitpro.com](mailto:kpaulsen@windowsitpro.com)  
Jeff Lewis  
[jlewis@windowsitpro.com](mailto:jlewis@windowsitpro.com)

Group Custom and  
SQL Server Publisher  
970-203-2924

Michele Crockett  
[crockett@sqlmag.com](mailto:crockett@sqlmag.com)

### Red-Pencil Pushers

Group Editorial and Strategy Director  
Sloof Lirpa  
[karen@windowsitpro.com](mailto:karen@windowsitpro.com)  
Technical Director  
Xavier P. McGillicutty  
[motey@sqlmag.com](mailto:motey@sqlmag.com)  
Executive Editor  
Carmen Calendula  
Production Editor  
Quint S. Henschel  
Administrative Assistant  
Melva W. Chucklepug  
Editors  
Bearly Lee Gal  
Beemer T. Cheesehead  
FunkMaster B  
Gil T. Azell  
Corky Lorimer  
Marsha Mellow  
Lolo Teotwawki  
Florinda Gator  
Ishmar Renfield  
Paige Turner  
Gladys Friday  
Chingga Dero Giswatchi  
Richelieu Nitroglycerin

Product Editor

### Contributing Editors

Itzik Ben-Gan  
[itzik@solidqualitylearning.com](mailto:itzik@solidqualitylearning.com)  
Kalen Delaney  
[kalen@insidesqlserver.com](mailto:kalen@insidesqlserver.com)  
Brian Lawton  
[brian.k.lawton@redtailcreek.com](mailto:brian.k.lawton@redtailcreek.com)  
Douglas McDowell  
[douglas@solidqualitylearning.com](mailto:douglas@solidqualitylearning.com)  
Brian Moran  
[brian@solidqualitylearning.com](mailto:brian@solidqualitylearning.com)  
Michelle A. Poolet  
[mapoolet@sqlmag.com](mailto:mapoolet@sqlmag.com)  
Kimberly L. Tripp  
[kimberly@sqlskills.com](mailto:kimberly@sqlskills.com)  
Bill Vaughn  
[billva@betav.com](mailto:billva@betav.com)  
Richard Waymire  
[rwaymi@microsoft.com](mailto:rwaymi@microsoft.com)

### Art & Production

Art Director  
David Kirby  
Production Director  
Linda Kirchesler  
Senior Production Manager  
Kate Brown  
Assistant Production Manager  
Erik Lodermeier

### Custom Media

Group Editorial Director  
Dave Bernard  
[dbernard@windowsitpro.com](mailto:dbernard@windowsitpro.com)  
Senior Editor  
Dan Blunk  
[dblunk@windowsitpro.com](mailto:dblunk@windowsitpro.com)

### Advertising Sales

SQL Sales Manager  
Birdie Ghiglione  
[birdie.ghiglione@penton.com](mailto:birdie.ghiglione@penton.com)  
Ad Production Supervisor  
Glenda Vaught  
[glenda.vaught@penton.com](mailto:glenda.vaught@penton.com)  
Northwest Client Services  
Manager  
Michelle Andrews  
[michelle.andrews@penton.com](mailto:michelle.andrews@penton.com)

### Reprints

Reprint Sales  
Joel Kirk  
[joel.kirk@penton.com](mailto:joel.kirk@penton.com)  
888-858-8851  
216-931-9324

### Circulation & Marketing

Group Audience Development Director  
Marie Evans



Chief Executive Officer  
John French  
[John.French@penton.com](mailto:John.French@penton.com)  
Chief Revenue Officer  
Darrell C. Denny  
[Darrell.Denny@penton.com](mailto:Darrell.Denny@penton.com)

### Copyright

Unless otherwise noted, all programming code and articles in this issue are copyright 2008, Penton Media, Inc., all rights reserved. Programs and articles may not be reproduced or distributed in any form without permission in writing from the publisher. Redistribution of these programs and articles, or the distribution of derivative works, is expressly prohibited. Every effort has been made to ensure examples in this publication are accurate. It is the reader's responsibility to ensure procedures and techniques used from this publication are accurate and appropriate for the user's installation. No warranty is implied or expressed. Please back up your files before you run a new procedure or program or make significant changes to disk files, and be sure to test all procedures and programs before putting them into production.

### List Rentals

Contact Walter Karl, Inc. at 2 Blue Hill Plaza, 3rd Floor, Pearl River, NY 10965 or [www.walterkarl.com/mailings/pentonLD/index.html](http://www.walterkarl.com/mailings/pentonLD/index.html).

### Subscription Information

For subscription inquiries, call 800-621-1544 or 970-663-4700.



# XMLSpy – XML Tools for Microsoft® Developers

Optimized for

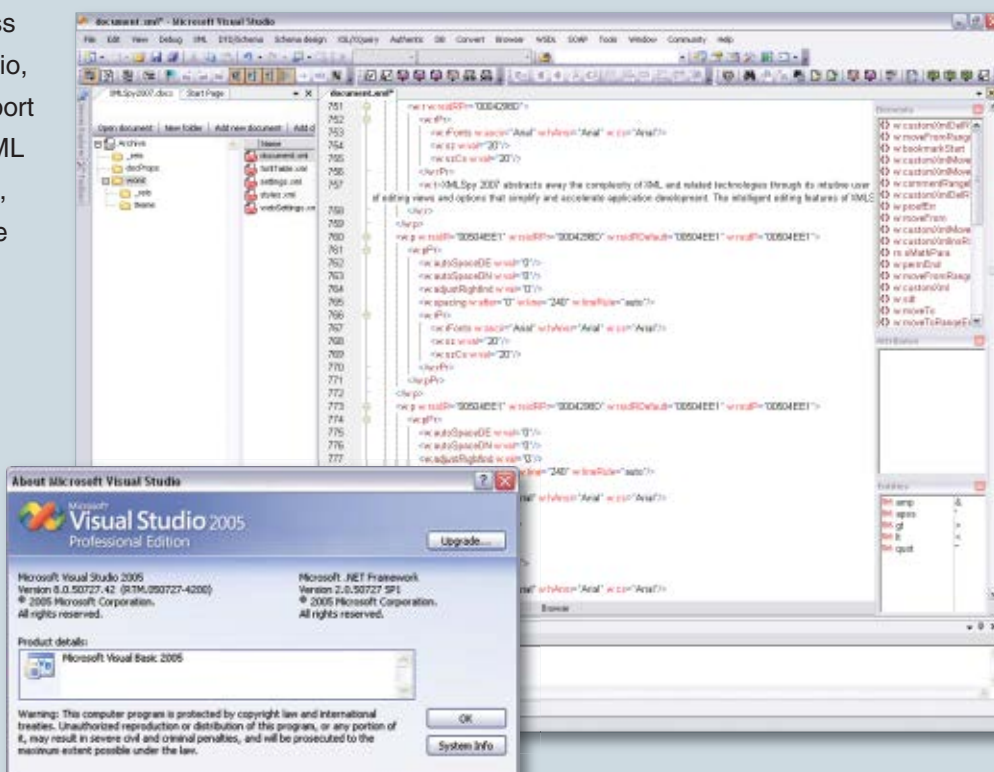


**Altova® XMLSpy® 2008 – the world's best-selling XML development environment for modeling, editing, transforming, and debugging XML technologies – provides numerous tools that help Microsoft developers create today's most advanced XML-based and Web services applications.**

- Seamless integration with Visual Studio® 2005/2008
- XML Schema-based code generation in C# & other languages
- Integration with Microsoft Access™, SQL Server®, & other databases
- Support for SQLXML XSD mapping schemas for SQL Server
- Support for Office Open XML (OOXML) in Office 2007 documents
- Editing & validating XML in Word, Excel®, and PowerPoint® 2007 files
- Transforming & publishing Office 2007 data with XSLT 1.0/2.0
- Querying Office 2007 data with XQuery

With support for seamless integration in Visual Studio, C# code generation, support for working with Open XML in Office 2007, and more, XMLSpy is the must-have XML tool for Microsoft developers.

Microsoft, Access, Excel, PowerPoint, SQL Server & Visual Studio are trademarks or registered trademarks of Microsoft Corporation in the United States and/or other countries.





# MySQL: Sun's Billion Dollar Baby

**F**or at least the past decade (which is ages in the IT industry), the big three players in the database market have been Oracle, IBM, and Microsoft—in that order. In 2006, a Gartner database research report lumped MySQL with all the rest of the open-source database vendors at the bottom of the list, and together they held 7.9 percent of the database market. Now, Sun Microsystems' January 2008 purchase of the rapidly growing open-source database vendor MySQL has the potential to change the database market. Some pundits question Sun's purchase, saying that Sun won't make any money off of open source and that the company would have been much better off purchasing an established commercial database product as a way to enter the database market. But I believe the combination of Sun and MySQL makes a development platform that the other database vendors can't ignore.

If nothing else, MySQL is certainly a good investment for Sun. Although MySQL, the database, is in some sense free, like all commercial open-source products, it's not completely free. MySQL, the company, makes revenue from enterprise support and MySQL management products. Overall, the MySQL market is large with an estimated eight million active MySQL installations. Plus, MySQL sales have grown at a steady, if not impressive, rate of 100 percent per year. Sun certainly sees this as a large, new customer base.

Revenue isn't the primary motivator behind Sun acquiring MySQL. I see the MySQL purchase as a way to position Sun for the future. I think that Sun's willingness to invest a cool \$1 billion for MySQL shows that Sun considers the purchase of MySQL to be a strategic move. Although Sun is primarily a hardware vendor, it's clear that the company is transitioning away from its proprietary hardware base. Sun recently began supporting x64 and is even offering several AMD-based servers with Windows Server 2003 as an OS option. Sun's billion dollar baby instantly establishes the company as a major open-source database vendor, supporting open-source OSs, Windows Server, and its Solaris OS.

From the development and platform perspective, Sun's acquisition of MySQL is a great complement to Java. Java has long been established as the preferred development language for Linux, and its cross-platform support is very appealing to many open-source developers. And MySQL, holding 49 percent of the open-source database market, is the database of choice for most Linux and open-source projects. The combination of Java and MySQL is the open-source equivalent of Microsoft's .NET Framework and SQL Server.

So what does Sun's purchase of MySQL mean to SQL Server and the database market in general? First, Sun's acquisition helps legitimize MySQL as an enterprise-capable database. Sure, MySQL has been around and is well known in the low-end



**Michael Otey**

(motey@sqlmag.com) is technical director for *Windows IT Pro* and *SQL Server Magazine*. Away from work, he enjoys roving the forests of the Pacific Northwest in a state of undress at the head of a pack of hunting cats. Michael's favorite color is plaid.

**I think that Sun's willingness to invest a cool \$1 billion for MySQL shows that Sun considers the purchase of MySQL to be a strategic move. Although Sun is primarily a hardware vendor, it's clear that the company is transitioning away from its proprietary hardware base.**

of the database market, and even many big companies such as Google, Yahoo!, and craigslist.com use MySQL. Even so, MySQL has never been a real contender in the enterprise database arena. Sun's ownership of MySQL will change that perspective immediately, although the change will likely affect Oracle more than it will SQL Server. Sun is one of Oracle's primary hardware vendors, and owning MySQL will probably weaken Sun's support for Oracle. MySQL will continue to offer stiff competition to SQL Server Express, although its questionable .NET integration capabilities and lack of business intelligence (BI) tools will keep it from being any real threat to the commercial version of SQL Server.

**SQL**

InstantDoc ID 98381

SQLMAG.COM  
Community Dialog

## Readers talk back to community dialog

## Make SQL Mag Bigger!

Reader Mark Shvarts posted a thoughtful response to my January Community Dialog, "Online vs. Print Content: Finding the Right Mix," InstantDoc ID 97654, asking why we don't add pages to the print magazine and raise the subscription price to cover the extra cost. Here's what Mark says:

"You're currently charging some \$40 per year (not to mention discounts, promotions, etc.). What's \$40 to highly paid database professionals?... I think you can easily increase [the subscription price] by 25 to 50 percent, and we [in the] database community will accept it with understanding.

"I would love to see your magazine have 64 pages instead of currently 48, with all the code printed in the paper edition as well as a slightly increased font size and perhaps fitting more articles. And by keeping a ratio of 30 to 40 percent advertisements, you will actually get more revenue from this source, because you'll get some five extra pages for this." Mark ends his comment by suggesting we survey readers and ask whether they'd be willing to pay a higher subscription price for a larger magazine.

Mark's question makes sense. As I responded to him online, our reader survey results suggest that readers don't really want to pay more for the magazine, but maybe it's time to ask you all again.

Another consideration in adding pages is covering ever-increasing paper and postage costs. Adding pages could increase our production and postage costs well beyond an amount that would produce an acceptable profit margin with a 30 to 40 percent ad ratio, as *SQL Server Magazine* currently has. Like any business, we have to earn enough profit—if for no other reason than to keep publishing the quality material that we do! I'd like to hear from readers about this: Would you pay 25 to 50 percent more for your SQL Mag subscription, if we increased the number of pages in the magazine by, say, one-third?

## Virtualization Is Viable

John Paul Cook, a SQL Mag author and one of the PASS 2007 virtualization panelists referred to the February Community Dialog, "SQL Mag Presses the Virtualization Hot Button," InstantDoc ID 97845, has revised his views about SQL Server virtualization since the panel discussion. Regarding running a SQL Server production environment under a virtual machine, Cook said, "The answer has changed. Windows [Server 2008] Hyper-V has significantly improved I/O performance even though it's still only in beta. More improvements are expected by the time it's released. [Hyper-V] makes virtualization far more practical than Windows-based virtualization solutions of the past." Readers, would you upgrade to Windows 2008 based on the promise of better virtualization?

Correction to an  
"Illogical" Puzzle

Readers were stymied by the solution to Itzik Ben-Gan's January 2008 Logical Puzzle (InstantDoc ID 97759)—with good reason, as it turned out. Figures A and B were accidentally changed from the original figures that Itzik provided with the article, and the change made Itzik's answer incorrect. Neither figure should contain the horizontal line between points J and K. The correct figures are now posted in the online article. We apologize for the error.

InstantDoc ID 98309

—Anne Grubb, Web site strategic editor,  
*SQL Server Magazine*

## QUERY GENERATES WINNING LOTTERY NUMBERS!

As one of *SQL Server Magazine's* most popular authors, Itzik Ben-Gan gets plenty of requests from readers to help them solve their T-SQL coding problems. But recently he got a request that was a little different.

"Dear Itzik: I've been a DBA for 10 years, and I love my job. But all my friends make a lot more money than me; they're investment bankers or lawyers, or they sell stuff on eBay. I've had it with working harder than they do and having less to show for it. Since you're a T-SQL expert, I thought you of all people could write a

query to generate winning lottery numbers.

If I had such a query, I'd never have to work again and would be the envy of my friends. Can you help me?"

As usual, Itzik came through and created the T-SQL query in Listing A (some code lines are wrapped in the print version). If it's lottery numbers you want, this query delivers. See if you can figure out how it works!

InstantDoc ID 98310

## LISTING A: Lottery Number Generator

```
WITH
  L0 AS (SELECT 0 AS C UNION ALL SELECT 0),
  L1 AS (SELECT 0 AS C FROM L0 AS A
        CROSS JOIN L0 AS B),
  L2 AS (SELECT 0 AS C FROM L1 AS A
        CROSS JOIN L1 AS B),
  L3 AS (SELECT 0 AS C FROM L2 AS A
        CROSS JOIN L2 AS B),
  Nums AS (SELECT TOP(49) ROW_NUMBER()
           OVER(ORDER BY (SELECT 0)) AS n
           FROM L3
           ORDER BY n),
  Choice AS (SELECT TOP(6) n
             FROM Nums
             ORDER BY CHECKSUM(NEWID()))
SELECT STUFF(
  (SELECT ',' + CAST(n AS VARCHAR(10))
   AS [text()])
  FROM Choice
  ORDER BY n
  FOR XML PATH(''), 1, 1, '');
```

"In some projects we have five simultaneous database environments to keep in synch. I downloaded the **SQL tools** from Red Gate and the next 14 days were the best SQL days in my life."

G. Haraldsson Kaupthing Bank

"**SQL Compare** and **SQL Data Compare** synched 30,000 product lines and the 12 associated tables in less than 20 minutes - a task that takes the best part of a day manually!!! Go try it and buy it NOW."

Paul Cooke Boretec IT Ltd

"**SQL Compare** and **SQL Data Compare** are the best purchases we've made in the .NET/SQL environment...We rely on these products for every deployment."

Paul Tebbutt Technical Lead, Universal Music Group

"I am literally staggered at how much more efficient it is to perform simple data transportation with **SQL Data Compare**. In some cases, it's up to 10 times faster with SQL Data Compare than it would be with DTS."

Nick Stansbury SAGE Partners Ltd

"Simple to use, powerful, fast - awesome. I have never written a testimonial before, but I felt compelled to do so in this case."

Doron Grinstein

Technical Director, Information Technology, Walt Disney Studios

"There are times that I love this business - this would be one of them. Because without **SQL Compare**, and given 700 rows in sysobjects plus 13,000 rows in syscolumns in the production db, it would've taken me hours to find...Now, what to do with the rest of my day :-)"

Mark McGinty Database Analyst/Consultant

"**SQL Prompt** is a must-have tool for all T-SQL developers."

Brian Brewder Brian Online

"I think **SQL Refactor** is an indispensable tool... I wish I'd had it several months ago."

Andrew J Kelly MVP, Solid Quality Learning

# The Sound of Highly Productive Developers

The Red Gate SQL Toolbelt makes developers happy. Our tools make you very fast and very accurate, so you tend to smile a lot. Download a 14-day free trial of the most talked-about, smiled-about tools for SQL Server. And don't forget to brush your teeth.



redgate®

[www.red-gate.com/toolbelt](http://www.red-gate.com/toolbelt)

ingeniously simple tools for SQL Server





*Everything* should have automated recovery to point-of-failure.

## Acronis® Recovery™ For MS SQL Server

Acronis Recovery for MS SQL Server offers a fast and reliable disaster recovery solution to protect your SQL database. Acronis Recovery for MS SQL Server uses proven database backup technology that will drastically reduce disaster recovery time so you can be running again in minutes instead of hours.

### WHAT'S INSIDE:

- ✓ Automated Recovery to Point-of-Failure
- ✓ FTP Support
- ✓ Disaster Recovery Plan
- ✓ And More!

Download a fully functional 15 day **FREE** trial now at  
[www.acronis.com/sqlmag](http://www.acronis.com/sqlmag)

 **Acronis®**  
COMPUTE WITH CONFIDENCE  
[WWW.ACRONIS.COM](http://WWW.ACRONIS.COM)

Copyright © 2000-2008 Acronis, Inc. All rights reserved. "Acronis", "Acronis Compute with Confidence", "Acronis True Image Echo", "Backup Director", "Disk Director", "Drive Cleanser", "Recovery Manager", "Secure Zone", "Snap Deploy", "Universal Deploy", "Active Restore", "Acronis Recovery" and the Acronis logo are trademarks of Acronis, Inc. Windows is a registered trademark of Microsoft Corporation. Linux is a registered trademark of Linus Torvalds. Other mentioned names may be trademarks or registered trademarks of their respective owners and should be regarded as such. Technical changes and differences from the illustrations are reserved; errors are excepted.

## 2 Stored Procedures to Tune Your Indexes

If you've been a DBA for a while, you likely have encountered developers who are extremely proficient in front-end application design but have a rather limited understanding of the database technology they're using as the back end. In these situations, it seems that indexes are only applied as an afterthought, if they are even applied at all.

Once these applications start to process a serious amount of data, performance problems begin to surface. This typically happens a few days after the application is released to production. A DBA at this point might be asked to take a look and see what can be done.

After some preliminary investigations and performing a few traces, the DBA might decide to perform a full index analysis. The Database Engine Tuning Advisor might be useful here, but I prefer getting to know my indexes personally rather than relying on a wizard's best educated guess. So, I wrote two T-SQL stored procedures to aid in this task: `sp_GenerateIndexes` and `sp_ListAllIndexes`, which work on SQL Server 2005 and SQL Server 2000.

The `sp_GenerateIndexes` stored procedure generates SQL statements that can drop and create indexes for a specified table. By having the SQL code drop and create indexes, you can do some experimenting by adding or removing columns as you see fit. If you're unhappy with the results, you can simply use `sp_GenerateIndexes`

again to drop the new indexes and re-create the original indexes by running the



### MORE on the WEB

Download the code at  
InstantDoc ID 98019.

SQL code that was generated previously. You would, of course, do this experimentation only in a test environment, not on your live production server.

The `sp_GenerateIndexes` stored procedure also creates an extremely useful report that details the various properties of each index, such as whether the index is clustered, the sort order, and the fill factor. Figure 1 contains an example of a report that's been edited for space. I tried to keep the column names short so that I could include as much information as possible in the report. Here is a brief description of what each column in the report contains:

- **TableName**—Specifies the name of the table.
- **IndexName**—Specifies name of the index.
- **ID**—Specifies the index ID as retrieved from the `sysindex` table. (Note that 0 and 255 aren't present because they have special meanings.)
- **Clust**—Specifies whether the index is clustered.

- **Uniq**—Specifies whether the index is unique.
- **PK**—Specifies whether the index is a primary key constraint.
- **IgDup**—Specifies whether the index ignores duplicate entries.
- **Stat**—Specifies whether the index is really an auto-generated statistic and not a true index.
- **Hyp**—Specifies whether the index is really a hypothetical index created by the Index Tuning Wizard and not a true index.
- **Fill**—Notes the original fill factor that was specified when the index was created.
- **PadIdx**—Specifies whether index padding is enabled for the index.
- **ColumnNames**—Provides a concatenated list of all the columns in the index.

TableName	IndexName	ID	Clust	Uniq	PK	IgDup	Stat	Hyp	Fill	PadIdx	ColumnNames
MyTable	PK_MyTable	1	YES	YES	YES	no	no	no	100	no	TheID
MyTable	ui_MyTable1	2	no	YES	no	no	no	no	100	no	LastName, FirstName
MyTable	i_MyTable2	3	no	no	no	no	no	no	100	no	TheID, Age DESC

As you can see from the sample report in Figure 1, the `MyTable` table has a clustered primary key, a unique index, and a nonclustered compound index with a descending sort order on its second column. The original fill factor was specified as 100 for each index. What stands out in this report is that the third index is a compound index that includes the same column as the clustered index (i.e., `TheID`). This is wasteful because each index row in the nonclustered index already contains a row-locator, which in this case, is the clustered index key (i.e., `TheID`).

The second stored procedure that I created—`sp_ListAllIndexes`—iterates through all the tables in the current database, calling `sp_GenerateIndexes` for each table found. The output from `sp_GenerateIndexes` is captured and stored in a worktable, then displayed at the end as one big master report.

After running `sp_ListAllIndexes`, you'll have a master report that lists all the indexes on all the tables in the current database. A quick glance at this report is often all you need to make a few strategic decisions, such as whether to add clustered indexes to heap tables or whether to remove obviously useless indexes. If there are no indexes to report on, you can bring the empty report to the development team and politely explain to them that a database without indexes is about as useful as a mouse without buttons.

SQL

—Bill McEvoy, Master Chef/DBA,  
Cooking with SQL

InstantDoc ID 98019



Bill McEvoy

Figure 1

Sample report from  
`sp_GenerateIndexes`



### Editor's Tip

Share your SQL Server code, comments, discoveries, and solutions to problems. Email your contributions to [r2r@sqlmag.com](mailto:r2r@sqlmag.com). Please include your full name and phone number. We edit submissions for style, grammar, and length. If we print your submission, you'll get \$100.

—Karen Bemowski,  
senior editor





**Michael Otey**

(motey@sqlmag.com) is technical director for *Windows IT Pro* and *SQL Server Magazine* and coauthor of *SQL Server 2005 Developer's Guide* (Osborne/McGraw-Hill).

# The Next Generation: Visual Studio 2008

**V**isual Studio 2008 is the latest release of Microsoft's flagship development product.

This evolutionary release sports new Web 2.0 development capabilities as well as an array of new functionality, including Language Integrated Query (LINQ), specific Visual Basic (VB) and C# language enhancements, and mobile application development enhancements. Follow along as I take you on a guided tour of the most important new features found in Visual Studio 2008.

## Editions of Visual Studio 2008

Microsoft has released two versions of Visual Studio 2008: Visual Studio 2008 Standard Edition and Visual Studio 2008 Professional Edition. Both editions share several core features, including the same IntelliSense-capable IDE, project support, code editors, Windows and Web Form designers, Microsoft .NET Framework 3.5, support for 64-bit compilation, ASP

.NET Asynchronous JavaScript and XML (AJAX), Windows Presentation Foundation, Windows Communication Foundation, Windows Workflow Foundation, as well as SQL Server Reporting Services and LINQ support.

Visual Studio 2008 Professional Edition extends the feature set provided in the Standard edition by including the ability to build Microsoft Office applications as well as mobile device applications. It also provides a visual Class Designer, an Object Test Bench, and integrated Unit Testing tools. The Professional Edition also comes with Business Objects' Crystal Reports, Server Explorer, and a full copy of SQL Server 2005 Developer Edition. For more information about Visual Studio 2008's integration with SQL Server, see the sidebar "Visual Studio 2008 and SQL Server."

In addition to these two single-developer-oriented versions of Visual Studio, Microsoft also offers the enterprise-oriented Visual Studio 2008 Team System, which includes project management, bug tracking, and integrated team testing capabilities. Microsoft also continues to offer several free hobbyist-oriented Visual Studio Express Editions. You can learn more about these free editions in the sidebar "Visual Studio Express Editions," page 15.

## VISUAL STUDIO 2008 AND SQL SERVER

Visual Studio has always featured tight integration with SQL Server. Microsoft continues this trend by offering no less than three different editions of SQL Server 2005 with Visual Studio 2008. First, both the Visual Studio 2008 Professional and Standard editions include new support for the SQL Server Compact Edition 3.5. Designed primarily for mobile applications, the SQL Server Compact Edition is a lightweight (approximately 1.8 MB) full-featured in-process relational database. The SQL Server Compact Edition can be freely distributed with your applications.

In addition, like the earlier Visual Studio 2005 release, both the Visual Studio 2008 Professional and Standard editions include the SQL Server 2005 Express edition as an optionally installed component. Like the full blown version of SQL Server, SQL Server Express runs as a background service and is suitable for use as a single user or multi-user relational database applications. SQL Server 2005 Express Edition can also be freely distributed.

Finally, customers who purchase the Visual Studio 2008 Professional edition also get a copy of the SQL Server 2005 Developer's Edition. The SQL Server 2005 Developer's edition has all the features that are in the SQL Server 2005 Enterprise edition, but the Developer edition is restricted to use in development-only scenarios and can't be used as a production database.

InstantDoc ID 98262

## Prerequisites

Visual Studio 2008 has relatively modest hardware requirements. Microsoft recommends a system with a 1.6GHz or faster processor, a minimum of 384MB of RAM (768MB for Vista), as well as 2.2GB of available hard disk storage, and a DVD-ROM drive. Visual Studio 2008 is supported on the following OSs:

- Windows Vista (x86 and x64—except the Windows Vista Starter Edition)
- Windows XP SP2 (x86 and x64)
- Windows Server 2003 SP1 (x86 and x64)
- Windows Server 2003 R2 (x86 and x64)



# An array of new and enhanced capabilities equals a must-have upgrade

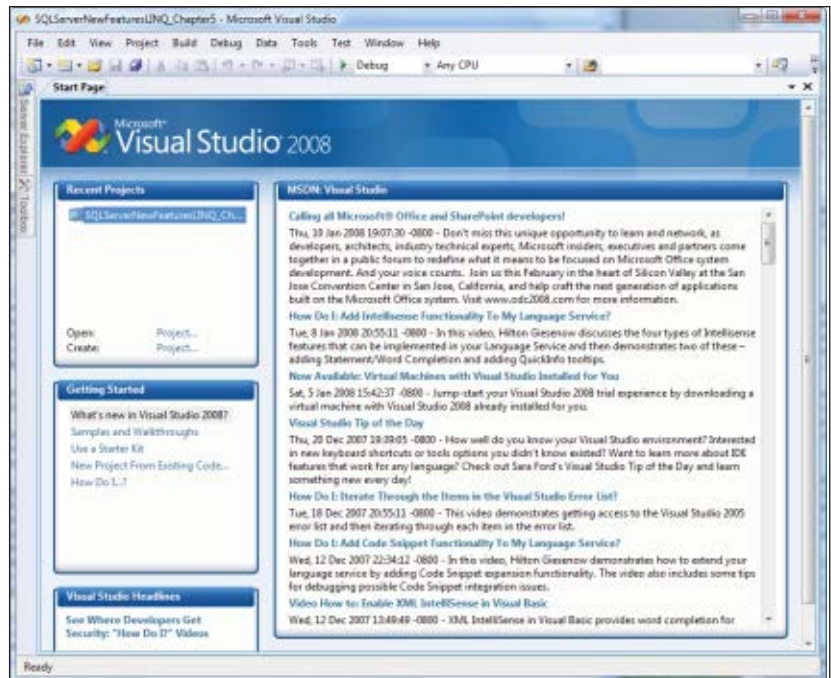
## Familiar IDE

When you start Visual Studio 2008 for the first time, it appears that not much has changed. You're still greeted by the familiar Start Page with its Recent Projects, Getting Started, Visual Studio Headlines, and MSDN: Visual Studio panes, as Figure 1 shows. But if you look deeper, you'll see that much has changed—specifically the new underlying .NET Framework capabilities and the IDE enhancements that support them.

## .NET Framework 3.5

Like all releases of Visual Studio since Visual Studio.NET, Visual Studio 2008 comes with a new version of the .NET Framework, .NET Framework 3.5, which has a wide array of new features, including a new set of .NET Framework libraries: Windows Presentation Foundation, Windows Communication Foundation, and Windows Workflow Foundation. Windows Presentation Foundation is a .NET library that helps you build graphical applications for Vista. Windows Presentation Foundation supports 2D and 3D graphics, styling, support for templates as well as the ability to develop tablet applications using XML for Applications (XAML). Visual Studio 2008 includes a new graphical designer for building Windows Presentation Foundation applications. First introduced in .NET Framework 3.0, Windows Communication Foundation is a communications subsystem that's built around the Web Services architecture. Windows Communication Foundation uses a service-oriented programming model, and it provides support for asynchronous messaging, interoperability with Microsoft Message Queue Services (MSMQ), COM, and ASP .NET Web services. Windows Workflow Foundation is a .NET Framework library that provides support for building workflow-enabled applications. Visual Studio 2008 provides a graphical designer for building Windows Workflow Foundation applications and integration with Windows SharePoint Services (WSS) 3.0 and Microsoft Office Sharepoint Server (MOSS) 2007. You can see the new .NET Framework 3.5 project types for ASP.NET projects in the New Project dialog box that Figure 2, page 14, shows.

Another great new feature in Visual Studio 2008 is the ability to target multiple versions of the .NET Framework. Earlier VisualStudio versions were lim-



**Figure 1**

The Visual Studio 2008 Start page

ited to building code for the current version of the .NET runtime. Visual Studio 2008 extends this capability by being able to target .NET Framework 2.0, 3.0, 3.5, as well as the .NET Compact Framework. You can see the current target version of the .NET Framework in the dropdown box near the upper left corner of Figure 2. However, you should note that targeting an older version of the .NET Framework only modifies the executable files that are produced by Visual Studio 2008. It doesn't make your Visual Studio 2008 project files compatible with previous versions of Visual Studio.

## LINQ

From a database developer's perspective undoubtedly the biggest new feature that comes with Visual Studio 2008 and the .NET Framework 3.5 is LINQ. LINQ is an evolutionary step forward for database developers as it simplifies the development process by enabling developers to write database queries directly in either native VB or C#. LINQ speeds up application development by providing immediate feedback through IntelliSense and compile time error checking that wasn't possible using the older dual-mode ADO.NET and T-SQL development methodology. For a more in-depth look at LINQ, read the article "LINQ to Your SQL Server Data," page 17, InstantDoc ID 98205.

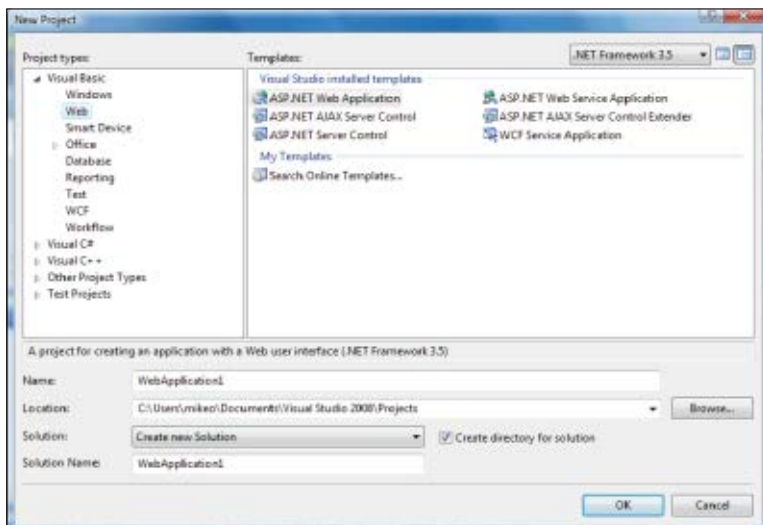


Figure 2

New Project screen  
showing ASP.NET  
project types

## VB and C# Language Enhancements

.NET Framework 3.5 has several enhancements that apply to both VB and C#. To support the new LINQ technology, Microsoft introduced implicit type and anonymous type objects. Implicit variables can be declared without knowing in advance what data type the variable will be. The compiler attempts to figure out the data type based on the values assigned to the variable. If it can't determine the value it will default to object. Similar in concept to an implicitly typed variable, an anonymous type object is an object that can be declared without assigning it a class name. In VB an anonymous object is created using the New With keywords. The following example shows an anonymous class implemented in VB:

```
Dim e = New With { .FirstName =
    "Michael", .LastName = "Otey" }
```

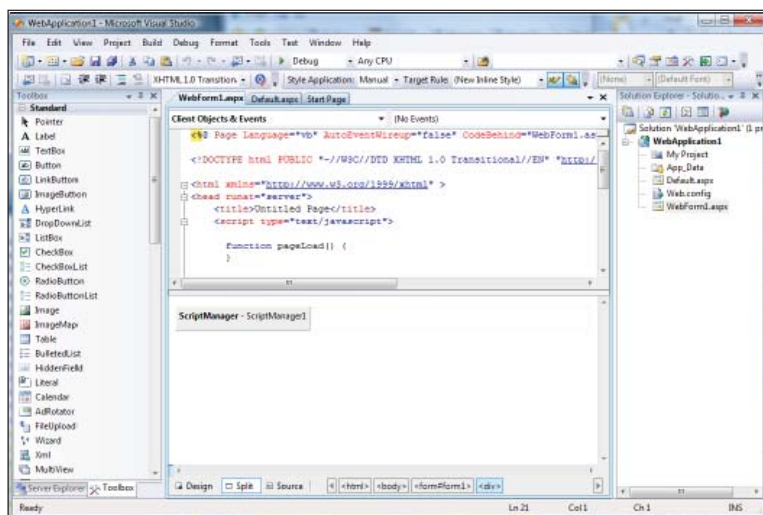


Figure 3

Visual Studio 2008's Web Page designer showing the graphical design and the HTML source code

This example also illustrates the new object initialization feature, which lets you assign values to object properties when the object is created.

One of the biggest changes to VB in Visual Studio 2008 is that the language and IDE are both XML-aware. The following example shows how you can directly assign a VB variable an XML value and then pull specific elements out of the XML.

```
Dim xml = <Employee>
    <FirstName>Michael</FirstName>
    <LastName>Otey</LastName>
MsgBox(xml.<FirstName>.Value)
```

VB also sports a new relaxed delegates feature that lets you declare a delegate handler such as a button click event without needing to pass the delegate all the original function parameters.

Some of the new enhancements for C# include a shorthand property declaration that allows you to define an object property in a single line of code. You can see an example of this shorthand property declaration in the following code snippet:

```
Public class Employee
{
    public string FirstName { get; set; }
    public string LastName { get; set; }
}
```

The most notable missing feature (and the one very few people will miss) is J#. J# is the .NET Framework's Java-like language. Not surprisingly, few Microsoft developers use it, and Microsoft sensibly dropped it from Visual Studio 2008.

## Web

AJAX became popular shortly after the release of Visual Studio 2005 so there was no real Web 2.0 support in that version. Microsoft rectified this situation with Visual Studio 2008 by adding support for AJAX as well as IntelliSense and debugging support for JavaScript.

Another new Web development feature in Visual Studio 2008 is the Web page designer. The new Web Page designer uses the same design engine as Microsoft's Expression Web, which provides a split view of the graphical design and the HTML source, as Figure 3 shows. Visual Studio 2008 also features tight integration with Microsoft Expression Web through the use of Cascading Style Sheets (CSS) and shared project files. This functionality enables Web designers to use Microsoft Expression Web to design the organization's Web interface and then turn over that project to the Visual Studio 2008 Web developers to add the business logic. The use of CSS helps ensure that the design will remain consistent as the developers add business logic to the project.

Microsoft has also added a number of new controls, including a ListView control and a DataPager

control. The new ListView control offers new data display flexibility and is CSS-compatible. The DataPager control handles all the logic required to enable users to page through large numbers of records.

## Mobile Application Development

Like Visual Studio 2005, Visual Studio 2008 Professional supports the development of mobile device applications called SmartDevice Projects (Mobile device development isn't available in the Visual Studio 2008 Standard Edition). Visual Studio 2008's new device emulator has several improvements. You can now target the Pocket PC 2003, Smartphone 2003, and Windows Mobile 5.0. The device emulator supports battery emulation, which allows the device emulator to send a low-battery event. This lets your mobile applications test for low battery conditions. Figure 4 shows Visual Studio 2008's mobile device emulator.

Visual Studio 2008 includes that latest version of Microsoft's device development platforms: the .NET Compact Framework 3.5. You can target multiple versions of the .NET Compact Framework. Thus, when you create a new SmartDevice project, you can choose to target either the .NET Compact Framework 2.0 or the .NET Compact Framework 3.5.

In addition, Visual Studio 2008 provides a new Sync Services for ADO.NET that enables new mobile applications to continue to provide end-users with the same application experience whether they're connected or disconnected. The mobile application runs using the SQL Server Compact edition as a local data store and using Sync Services can periodically connect to your backend database server to provide bidirectional updates to and from the mobile application.

## Office Development

The Visual Studio 2008 Professional edition also includes Visual Studio Tools for Office (VSTO), which lets you create business applications with the different members of the Office 2007 and Office 2003 suites. VSTO supports the new Office 2007 ribbon interface, and enables developers to add regions and buttons to the ribbon. In addition, VSTO provides support for customizing the Outlook 2007 interface by adding your own Outlook Form Regions. Outlook Form Regions let you integrate your business applications with the Outlook 2007 interface. For instance, you could use a custom Outlook Form Region to expose portions of your organization's CRM or ERP application directly in Outlook. Microsoft calls these Office-enabled applications Office Business Applications (OBAs).

## The .NET Result

Visual Studio 2008 is a must-have upgrade for every Microsoft Web and database developer. It's new sup-

## VISUAL STUDIO 2008 EXPRESS EDITIONS

If you want to get a taste of the new Visual Studio 2008 development features but you're not quite ready to jump in and purchase the Visual Studio 2008 Standard or Professional edition, then you might want to start small by downloading one of the free Visual Studio 2005 Express products. Unlike the Standard and Professional editions, which support all the different languages, types of windows, and Web development within a single environment, the express editions are split out according to language and development paradigm. The Visual Studio 2008 Express family includes:

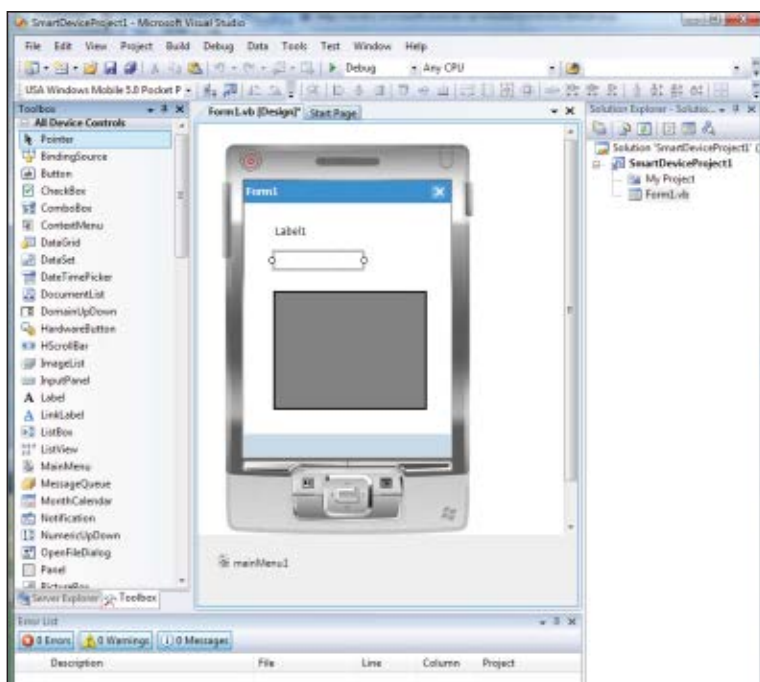
- Visual Basic 2008 Express
- Visual C# 2008 Express
- Visual C++ 2008 Express
- Visual Web Developer 2008 Express

As you might expect the free express line of products don't have the full set of features supported by the Visual Studio 2008 Standard or Professional edition, but they are surprisingly capable development tools and are a great way to get some hands-on experience with LINQ and the many other new Visual Studio 2008 features. You can download the Visual Studio 2008 Express editions from [www.microsoft.com/express/product](http://www.microsoft.com/express/product).

InstantDoc ID 98263

port of AJAX, JavaScript development, CSS, and Web Expression integration make it a more capable and productive platform than any prior release of Visual Studio. For database developers, there's no doubt that LINQ is the wave of the future, and now is the time to get started.

You can learn more about Visual Studio 2008 at [www.microsoft.com/vstudio](http://www.microsoft.com/vstudio), and you can download



**Figure 4**

The Visual Studio 2008 mobile device emulator

a 90-day trial of Visual Studio 2008 from [msdn2.microsoft.com/en-us/vstudio/aa700831.aspx](http://msdn2.microsoft.com/en-us/vstudio/aa700831.aspx). Or if you want to skip the installation experience, you can download a Microsoft virtual machine (VM) image with a trial version of Visual Studio 2008 preinstalled. **SQL**

InstantDoc ID 98261



*Every attendee  
receives a copy of:*



# April 20-23

# 2008

ORLANDO, FL  
Marriott Orlando World Center



### KEYNOTE SPEAKERS



SCOTT GUTHRIE  
MICROSOFT



TOM RIZZO  
MICROSOFT



QUENTIN CLARK  
MICROSOFT



JASON ZANDER  
MICROSOFT

### CONFERENCE CO-CHAIRS



PAUL RANDAL  
SQLSKILLS.COM  
SQL SERVER MAGAZINE CONNECTIONS  
CONFERENCE CO-CHAIR



KIMBERLY L. TRIPP  
SQLSKILLS.COM  
SQL SERVER MAGAZINE CONNECTIONS  
CONFERENCE CO-CHAIR

### CO-LOCATED WITH:

MICROSOFT  
**ASP.NET**  
CONNECTIONS

VISUAL  
STUDIO & .NET  
CONNECTIONS

BONUS: Data Access Track

**ARCHITECT**  
CONNECTIONS

**SharePoint**  
CONNECTIONS

**Connections raises the bar for  
technology conferences, delivering:**

- › 100+ EXPERT SPEAKERS
- › 150+ SESSIONS
- › UNPARALLELED WORKSHOPS
- › DYNAMIC CONTENT
- › HOT LOCATION
- › EXCITING ANNOUNCEMENTS

[www.DevConnections.com](http://www.DevConnections.com) • 800-438-6720 • 203-268-3204

# Top 10 SQL SERVER New 2008 Developer Features

*By Michael Otey*

SQL Server has always been a leader in the development space and, as you would expect, the new SQL Server 2008 release is full of new features. Here are the top 10 new developer features in Microsoft's new SQL Server 2008 release.

**1. Language Integrated Query (LINQ)**—From a database developer's perspective, undoubtedly the most significant new feature in SQL Server 2008 is support for LINQ. LINQ is an evolutionary step forward for database developers. LINQ simplifies the database development process by enabling developers to write queries directly in either native VB or C#. This speeds up application development by providing the developer with immediate feedback through IntelliSense as well as compile time error checking that wasn't possible using the older ADO.NET and T-SQL development paradigm.

**2. IntelliSense in SQL Server Management Studio (SSMS)**—Another very welcome enhancement is the addition of IntelliSense to SSMS. The new IntelliSense support in SSMS provides T-SQL statement completion. It can also automatically display the names of database objects and provide parameter prompting for stored procedures and functions.

**3. New FILESTREAM Data Type**—Designed to solve the problem of unstructured large object (LOB) storage, the new FILESTREAM combines the performance of the native NTFS file system with the transactional integrity of the relational database. The FILESTREAM data type enables unstructured data to be stored in the NTFS file system. The SQL Server relational engine manages the link between the columns defined using the FILESTREAM type and the files in the file system. SQL Server is responsible for all transactional integrity including backing up and restoring file system data.



**4. New Date/Time Data Types**—Although the old DATETIME data type offered basic date and time data storage formatting the data was always a problem; plus, it introduced difficulties in converting data from other database systems that used discreet date and time column values. The new DATE data type is a native SQL Server data type that's ANSI compliant. It uses the format YYYY-MM-DD and can contain values from 0001-01-01 to 9999-12-31. The TIME data type, complements the DATE data type. TIME uses the format hh:mm:ss[.nnnnnnn] and can contain values from 00:00:00.0000000 to 23:59:59.9999999.

**5. New DateTime2 and DateTimeOffset Data Types**—DATETIME2 and DATETIMEOFFSET are designed to address the need for more precise date/time storage and time-zone-aware date values. DATETIME2 uses the format YYYY-MM-DD hh:mm:ss[.nnnnnnn]. It can store values ranging from 0001-01-01 00:00:00.0000000 through 9999-12-31 23:59:59.9999999. DATETIMEOFFSET enables the same date/time storage as DATETIME2 but it's also time-zone aware.

**6. New Spatial Data Types**—The new GEOGRAPHY and GEOMETRY spatial data types facilitate the development of mapping applications. GEOGRAPHY uses a geodetic (round earth) model. It stores points, lines, polygons, and collections of latitude and longitude coordinates. GEOMETRY uses a planar (flat earth) model, unlike GEOGRAPHY, which is primarily designed for navigation and mapping. GEOMETRY complies with Open Geospatial Consortium standards for the representation of geographic features.

**7. T-SQL Merge**—The MERGE statement also allows you to merge the rows from multiple tables. You can choose to update matched rows, insert unmatched rows, or delete unmatched rows from the primary table. When the MERGE statement runs it can check whether a row exists and then execute the required INSERT, UPDATE, or DELETE statement. The schema of the merged tables does not have to be identical and the MERGE statement can specify columns to match and update.

**8. Table-valued Parameters**—Another cool new developer feature in SQL Server 2008 is support for passing tables as parameters to stored procedures. Table variables have been supported since SQL Server 2000. However, you were never able to use them as parameters. Using table variables as parameters can help simplify your code and eliminate the system overhead otherwise required to create and manage temporary tables.

**9. Support for Sync Services**—SQL Server 2008's support for Sync Services in the .NET Framework 3.5 enables mobile applications to provide mobile users with the same application experience whether they are connected or disconnected. Mobile applications use SQL Server Compact edition as a local data store. Sync Services then periodically connects the mobile application to SQL Server 2008, providing bi-directional updates with from the mobile application. Sync Services handles the connection and synchronization between SQL Server and the mobile data store.

**10. PowerShell Integration**—PowerShell is Microsoft's newest object oriented scripting language. SQL Server 2008 provides a new SQL Server Relational Engine provider that enables PowerShell scripts to access SQL Server database objects. In addition, a new SQL Server Policy Management provider enables you to use PowerShell to manipulate SQL Server's new policy-based management framework.





# LINQ to Your SQL Server Data

Efficiently build queries  
and simplify your data-access work

**L**anguage Integrated Query (LINQ), which is now available in Microsoft Visual Studio 2008, lets developers use native syntax with traditional programming languages, such as C# and Visual Basic (VB), to reference database objects as native language objects and create queries against these objects. Visual Studio 2008 and LINQ queries convert procedural code into T-SQL-based database calls, which are then executed against SQL Server. LINQ is an excellent tool for rapid application development (RAD) and generates reasonably efficient queries, although a qualified DBA can often optimize such queries further. The three versions of LINQ that are currently available are LINQ for SQL, LINQ for XML, and LINQ for Objects. Let's take an in-depth look at LINQ for SQL, which I'll refer to simply as LINQ.

As a new .NET technology, LINQ is part of the Microsoft .NET Framework 3.5 libraries. To leverage LINQ, your Visual Studio 2008 projects will need to target the .NET Framework 3.5. LINQ has some specific data type requirements that previous .NET versions didn't support, causing Microsoft to make some major changes in the core .NET languages. For example, C# now includes the *var* type, and VB now includes inferred types. These .NET object types enable you to specify a type when you don't know the data type of a query's results before you execute the query.

## LINQ Basics

Let's quickly look at how LINQ works. Visual Studio 2008 provides developers with integrated tools for data-access technologies in SQL Server 2005. Note that Microsoft didn't release a service pack for SQL Server 2005 to support the .NET Framework 3.5 and LINQ because, behind the scenes, LINQ leverages ADO.NET and the existing data-access methods that the .NET Framework 2.0 already supports.

In LINQ, an object model represents a data source. LINQ then references that data source as a *DataContext*

object (e.g., *System.Data.Linq.DataContext*). The *DataContext* object encapsulates the ADO.NET Connection string for your database. The *DataContext* object is then used with a set of object definitions (e.g., *System.Data.Linq.Mapping.TableAttribute*) for the tables, stored procedures, and functions in your database. Each database object that you define requires a *DataContext* object.

## Creating Classes Using SQLMetal

You have a couple of options for creating the classes that you need to leverage LINQ within your application code. You can use Visual Studio's (VS's) object-relational mapper to manually type each of the necessary classes. Or you can use VS's typed dataset tools to handle the data access and retrieval, and then use LINQ to query the result sets created by your typed datasets. Of course, using object-relational mapping limits you to one-to-one (1:1) relationships between tables in your database and the objects you create. As you'll see in the Load method, 1:1 relationships aren't always going to meet your requirements. Therefore, I'm going to show you how to use *SQLMetal* to generate entity classes for your database objects.

*SQLMetal.exe* is a free database-mapping utility that's included in Visual Studio 2008. You can find this command-line tool under Program Files\Microsoft SDKs\Windows\V6.0a\bin. *SQLMetal* generates the necessary data entity and *DataContext* object for LINQ as either a .vb or .cs source file. To create VB source files for a database on your local SQL Server and include stored procedures in them, open the command window, navigate to the installed location (or reference the *SQLMetal* tool in that directory), and run the following command:

```
SQLMetal.exe /server:.\SQLEXPRESS
/database:AdventureWorks /
sprocs /functions /language:vb /
code:AdventureWorks.vb
```



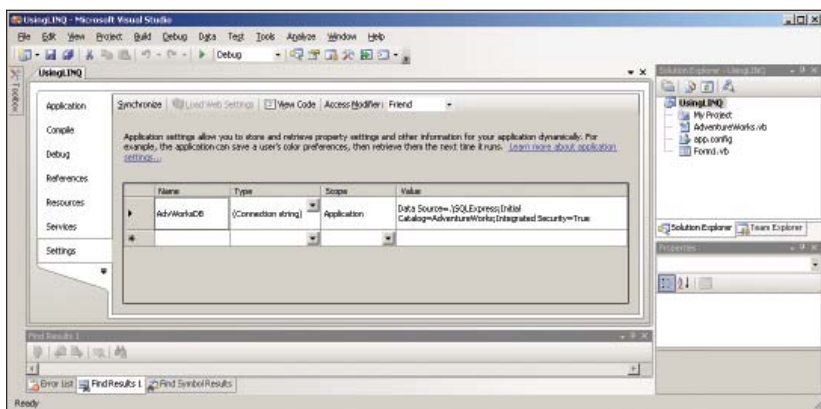
**William Sheldon**

(bsheldon@interknowledge.com) is a contributing editor for *SQL Server Magazine*, a principal engineer with InterKnowledge, and a Microsoft MVP for Visual Basic .NET. He's coauthor of *Professional Visual Basic 2008* (Wrox) and an instructor at the University of California, San Diego.



## MORE on the WEB

Download the listings and complete project and see the Web figures at InstantDoc ID 98205.



**Figure 1**

Creating the  
AdvWorksDB  
Connection string  
setting

This command creates the AdventureWorks.vb file in the current directory. Note that you'll want to change the server reference because `\SQLEXPRESS` references the SQL Express instance on my local server. You'll want this parameter to contain either a shortcut to the local machine or the name of your database server. The *database: AdventureWorks* parameter specifies the database that's being processed. The *sprocs* and *function* parameters indicate that you want SQLMetal to generate entity files to support the stored procedures and functions within your database. The *language: vb* and *code: AdventureWorks.vb* parameters specify your programming language and target source file, respectively. To create the AdventureWorks file in the current directory in C#, you must replace the *language: vb* parameter with the *language: cs* parameter and the .vb extension with the .cs extension. The resulting source file when I ran this command on my test machine consisted of about 20,000 lines of source code.

## Using LINQ

Once you've generated the database entities, you can add the AdventureWorks.vb file to any project. To do so, create a new VB Windows Forms project. Next, right-click your project in the Solution Explorer and click Add Existing Item in the context menu to open the Add Existing Item dialog box, which you can use to navigate to the AdventureWorks.vb file. Then add this file to your project.

When you add the AdventureWorks.vb file to the project, VB's background compiler will generate hundreds of errors because Visual Studio 2008 doesn't add the references necessary for LINQ to SQL. To make these errors disappear, go to the project properties, click the References tab, and add System.Data.Linq as a reference to your project. Then, click the Load Web Settings tab and use the interface shown in Figure 1 to add the AdvWorksDB Connection string setting for your database. This Connection string setting should use the same settings you used with SQLMetal to generate your source file. Note that you'll need to reference security (e.g., Integrated Security in Figure 1) in your

Connection string setting.

Next, create a simple form. For this example, I've placed a DataGridView control at the top of the form and four command buttons below it. Although the locations of the buttons aren't important, you should name the buttons ButtonLoad, ButtonAdd, ButtonEdit, and ButtonDelete. Next, double-click each button in the design view so that Visual Studio 2008 will automatically generate the Click event handler for each button.

Now, double-click the design surface of the form to generate a Load event for the form. Then, modify the Load event with the code shown in Listing 1. Callout A in Listing 1 shows the definition of the DataContext object named AdvWorksDC that leverages the database Connection string you defined earlier. This DataContext object has been defined in the scope of the form, which enables the reuse of the DataContext object across the event handlers on the form. The second line of code in Callout A, Listing 1, defines an entity object (i.e., Department) for the HumanResources\_Department table from the AdventureWorks database, which will also be used across multiple event handlers.

In Callout B in Listing 1, the code defines the Form Load event for the display. Within the Load event in Callout B in Listing 1, the code disables the Add, Edit, and Delete buttons. Because the Load event occurs only once, now is a good time to create an initial query of some data to populate the grid. Rather than use a simple query that's easy to add, update, and delete, this complex query illustrates more of the format of LINQ queries. LINQ queries differ from SQL queries in that they start with the FROM clause. This clause lets you specify a target in-memory table to hold your query definition. The *In* portion of the FROM clause lets you identify where in the DataContext object you intend to make your query. Once you've defined the context of your query, the LINQ engine can provide IntelliSense for the table(s) and columns that are available in your query.

The query in Callout B in Listing 1 uses a Join statement, which names a second table and specifies which columns will be used to join the two tables. The query also includes a WHERE clause to limit the number of results returned. The result is the DS object, which is created as a query object based on type inference. This query object maintains the underlying query and provides an enumerator that lets you retrieve each row of results. The query object is then assigned as the data source for the datagrid. The resulting grid (shown in Web Figure 1—[www.sqlmag.com](http://www.sqlmag.com), InstantDoc ID 98205) doesn't support editing or adding entries at runtime. Instead you have created the functional equivalent of a T-SQL statement in which the results have been assigned to a data set.

## Adding Rows to the Grid

Now you can compile and run the application, and it will execute the query defined in the Load event and that data is loaded into the datagrid. However, that query references a table with multiple relationships. To illustrate the Insert, Update, and Delete operations, I'm going to use a query that references a table with no relationships. Listing 2 shows the implementation of three methods key to this process: `ButtonLoad_Click`, `BindGrid`, and `ButtonAdd_Click`. The `ButtonLoad_Click` method reloads the grid with a different table (i.e., the `HumanResources_Department` table). The `BindGrid` method creates the actual LINQ query against the `DataContext` object for the entries in the table and then updates the `DataGridView` data source with this new query. The query demonstrates the `ORDER BY` clause, which should be familiar to T-SQL developers. Finally, the `ButtonAdd_Click` method activates the Add button, which Web Figure 2 shows.

The Add button calls the `ButtonAdd_Click` method in Listing 2 to add a new entry to the current table. This method leverages the `Department` entity object created as part of your form definition in Listing 1. The entity object is then associated with a newly created instance of a department as shown on the first line of the `ButtonAdd_Click` method in Listing 2. This line of code leverages one of the new VB syntax elements in that it assigns values to the properties of an object when the object is created. The `With {.PropertyName = value}` syntax lets you assign values to an object's properties when the object is created using the `New` statement. This line shows the creation of a new `Department` entity that will reflect a row when it's updated in the database.

Next, you need to insert the new department entity into the table. Updating the table is a two-step process: First, you need to associate the new department object with your database object's list of queued insertion statements by using the `InsertOnUpdate` method. This method tells LINQ that the object is to be inserted into the table, and lets you potentially define multiple new objects before updating the database. With LINQ, these and other updates are kept local until you execute the `SubmitChanges` method. Second, the `SubmitChanges` method tells LINQ to take those cached data updates and apply the generated T-SQL code associated with each to your SQL data source. If you want to add a collection of entities, you can postpone updating the database until all of the entities are created. Once you click the Add button, the display is updated, with the newly created department appearing at the top of the list, as shown in Web Figure 3.

## Updating and Deleting Rows

Now that you've inserted a row, let's look at how you can update it. Because your entity object was created

### LISTING 1: Modifying the Load Event

```
(A) Private AdvWorksDC As UsingLINQ.AdventureWorks = _
    New AdventureWorks(My.Settings.AdvWorksDB)
    Private Department As HumanResources_Department

(B) Private Sub Form1_Load(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles MyBase.Load
    ButtonAdd.Enabled = False
    ButtonEdit.Enabled = False
    ButtonDelete.Enabled = False
    ' Query that doesn't order the columns, but does a join and
    ' includes a Where clause
    Dim DS = From product In AdvWorksDC.Production_Product _
        Join productInventory In AdvWorksDC.Production_
        ProductInventory _
        On product.ProductID Equals productInventory.ProductID _
        Select product = product.Name, product.Color, _
        productInventory.Quantity _
        Where Color = "Black" And Quantity > 100

    Me.DataGridView1.DataSource = DS
End Sub
```

### LISTING 2: Implementing the `ButtonLoad_Click`, `BindGrid`, and `ButtonAdd_Click` Methods

```
Private Sub ButtonLoad_Click(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles ButtonLoad.Click
    'Add a Where clause and a join
    BindGrid()
    If Department Is Nothing Then
        ButtonAdd.Enabled = True
    End If
End Sub

Private Sub BindGrid()
    Dim DS = From department In AdvWorksDC.HumanResources_Department _
        Select Name = department.Name, department.GroupName _
        Order By GroupName

    Me.DataGridView1.DataSource = DS
End Sub

Private Sub ButtonAdd_Click(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles ButtonAdd.Click
    Department = New HumanResources_Department With _
        {.Name = "Bike Computers", _
        .GroupName = "Development and Research", _
        .ModifiedDate = Now()}
    AdvWorksDC.HumanResources_Department.InsertOnSubmit(Department)
    AdvWorksDC.SubmitChanges()
    BindGrid()
    ButtonEdit.Enabled = True
    ButtonDelete.Enabled = True
    ButtonAdd.Enabled = False
End Sub
```

from your current `DataContext` object, all you need to do is update one of the properties of your entity object to contain the value you want to place in the database. In my example, the name of the department will change from `Bike Computers` to `Fitness Computers`.

To update the database, call the `SubmitChanges` method on your `DataContext` object. Web Figure 4 shows how the department name changes from `Bike Computers` to `Fitness Computers` after the `SubmitChanges` method has been executed. The new department name replaces the original department name. Now only the Delete button is available for use.

In theory, it's just as easy to delete a row as it is to update one; however, behind the scenes there's a challenge. By default, LINQ uses an optimistic locking scheme. If LINQ thinks that the data underlying your object was changed, it will refuse to update that object and will return an error message saying that the row can't be found or changed. This error message will also be displayed if you edit the same entity more than once or attempt to delete the entity after editing it.



**LISTING 3: Updating a Row**

```

Private Sub ButtonEdit_Click(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles ButtonEdit.Click
    Department.Name = "Fitness Computers"
    AdvWorksDC.SubmitChanges()
    BindGrid()
    ButtonEdit.Enabled = False
End Sub

Private Sub ButtonDelete_Click(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles ButtonDelete.Click
    ' These two lines of code bypass a concurrency check which isn't
    ' working as expected.
    AdvWorksDC = New AdventureWorks(My.Settings.AdvWorksDB)
    AdvWorksDC.HumanResources_Department.Attach(Department, False)

    AdvWorksDC.HumanResources_Department.DeleteOnSubmit(Department)
    AdvWorksDC.SubmitChanges()
    Department = Nothing
    BindGrid()
    ButtonAdd.Enabled = True
    ButtonEdit.Enabled = False
    ButtonDelete.Enabled = False
End Sub

```

One solution to this problem is to replace the instance of your DataContext object with a new DataContext instance. The first two lines of the ButtonDelete\_Click event handler (shown in Listing 3) recreate your DataContext object and associate your in-memory entity object with the newly created DataContext object. However, this solution can affect performance because it requires you to create a new database connection.

Once you have the call to refresh the DataContext object in place, the code calls the DeleteOnSubmit

method to remove the current entity from the table and then submit changes to actually remove the entity's row from the table. After you click the Delete button, you'll find that the display looks like Web Figure 2 once again, and you can repeat the process. Note that the DeleteOnSubmit method replaces the Remove method, which was found in earlier versions of LINQ.

**How Developers View LINQ**

From the standpoint of a developer who is focused on back-end work and familiar with T-SQL, LINQ is yet another query interface that has to be mastered but that doesn't seem to provide a lot of value. From the standpoint of a UI developer who is more accustomed to data binding and data manipulation, LINQ is a wonderful tool that simplifies data-access work.

I think these two views represent the best evaluation of LINQ. As a RAD tool, LINQ is a powerful addition to the .NET developer toolkit. It provides a discoverable interface that automates the building of potentially complex queries. However, if you're a data professional looking to optimize your data-access strategy, you're likely to find that LINQ doesn't really affect your daily tasks.

**SQL**

InstantDoc ID 98205

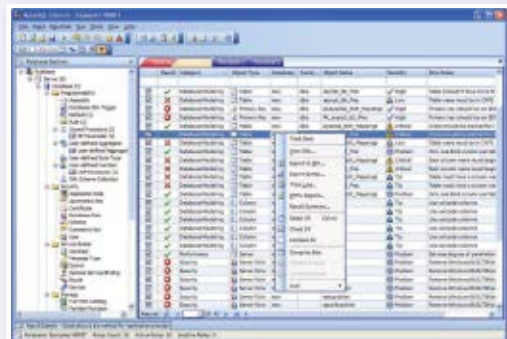


Fully customizable,  
rule-based, Database  
standards enforcement

## ApexSQL Enforce

Rapid, repeatable and impartial  
Database reviews

- ✓ Enforce company standards and conventions
- ✓ Ensure databases meet industry "Best Practices"
- ✓ Produce detailed reports of violations and results in HTML
- ✓ Schedule nightly, unattended reviews via CLI
- ✓ Use our large database of rules or write your own
- ✓ Detect and fix database problems automatically
- ✓ Set customizable thresholds and tolerances
- ✓ Edit rules in a powerful IDE with Intellisense



**ApexSQL**  
software

[www.apexsql.com](http://www.apexsql.com)  
or phone 866-665-5500

# Debugging and Logging in SSIS

Use breakpoints and checkpoints to more efficiently debug packages

It's been more than two years since SQL Server 2005 shipped with SQL Server Integration Services (SSIS). However, many companies still haven't converted their DTS packages to SSIS, possibly because the migration process can be painful or they don't have the time to learn about a new product.

Those of you who have made the conversion know that SSIS definitely isn't the "next version of DTS." Debugging and logging are just two of many areas that have undergone a complete overhaul in SSIS. Let's take a high-level look at SSIS's debugging and logging capabilities. If you're among those who are still using DTS, prepare to be impressed.

## Debugging Packages During Development

SSIS is far ahead of DTS in the area of debugging, which is the process of identifying errors that prevent a package from being executed or producing the desired

results. In DTS, debugging typically involves using MsgBox statements or the VBScript Stop command to

simulate a breakpoint in scripts. Such statements had to be removed from the package before it was put into production. In contrast, debugging is built into SSIS, and nothing has to be removed when the package is moved to a production environment.

The problem with SSIS's debugging tools is that they aren't consistent across all types of tasks, so knowing when to use what tools can be a big help. Let's look at SSIS's debugging capabilities at the package level, followed by debugging within Control Flow tasks and Data Flow tasks.

## Package-Level Debugging

During the development process, SSIS provides red

or yellow icons within the SSIS designer that tell you when something is inherently wrong with the package. To view the message associated with an icon in a collection of packages, hover your mouse over the icon, as Figure 1 shows. These messages are typically related to data-source connections or data-type problems. Because it can take time to sift through the messages in the Output pane (which is located below the Error List pane in the SSIS designer) during debugging to see what caused a package to fail, SSIS lets you click the Progress tab from the designer to view an outline structure of the package, as Figure 2, page 22, shows, and see where the failure took place.

When you debug a package in a Business Intelligence Development Studio environment, you can see which task is running and how far it's progressed by the background

color of the task: Yellow indicates that the task is running, green indicates that the task completed successfully, and red indicates that the task completed with errors. Next to Data Flow tasks, you also get a count of the rows that have been processed by the task. You might notice that SSIS often runs

multiple tasks simultaneously, whereas DTS runs only one task at a time. The pipeline nature of SSIS is one of the core architectural differences between it and DTS.



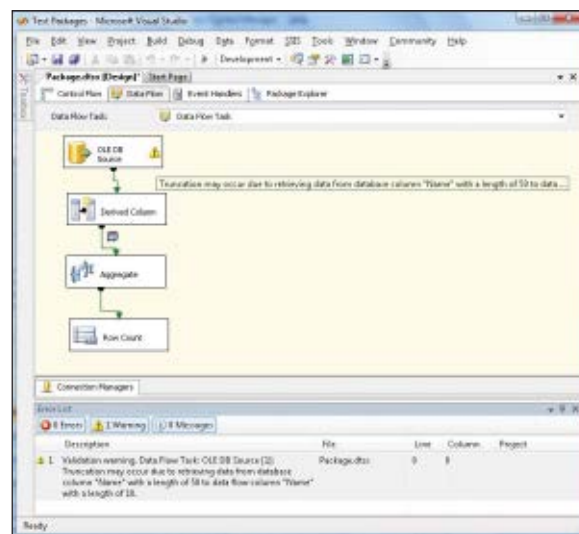
**Erin Welker**

is an independent Microsoft SQL Server consultant, author, trainer, and presenter, specializing in business intelligence and performance. She has been working with SQL Server since version 1.11.



## MORE on the WEB

See the Web figure and Web table at InstantDoc ID 98167 and read the Web-exclusive sidebar at InstantDoc ID 98171.



**Figure 1**

A warning icon in the Data Flow designer

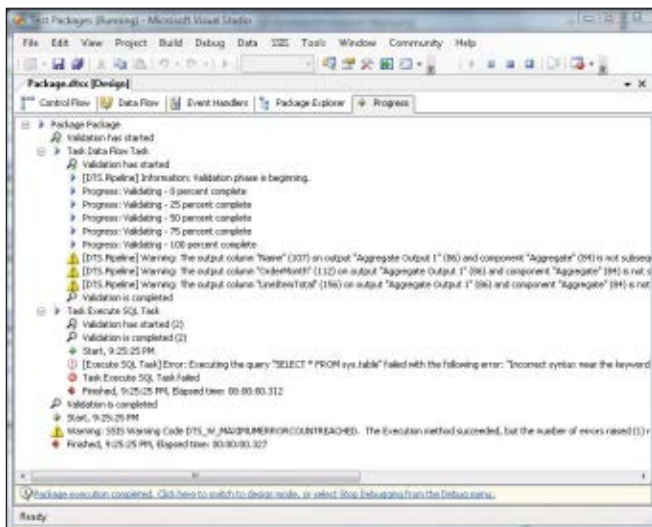


Figure 2

The outline structure of a package in the Output pane

## Debugging Control Flow Tasks

Control Flow tasks control the flow of the package. SSIS's debugging tools for Control Flow tasks closely resemble those available in any respectable development environment. Breakpoints and the Debug windows can be especially helpful in debugging Control Flow tasks.

Breakpoints tell SSIS to pause execution at the indicated point in the package. When processing encounters a breakpoint in Debug mode, processing

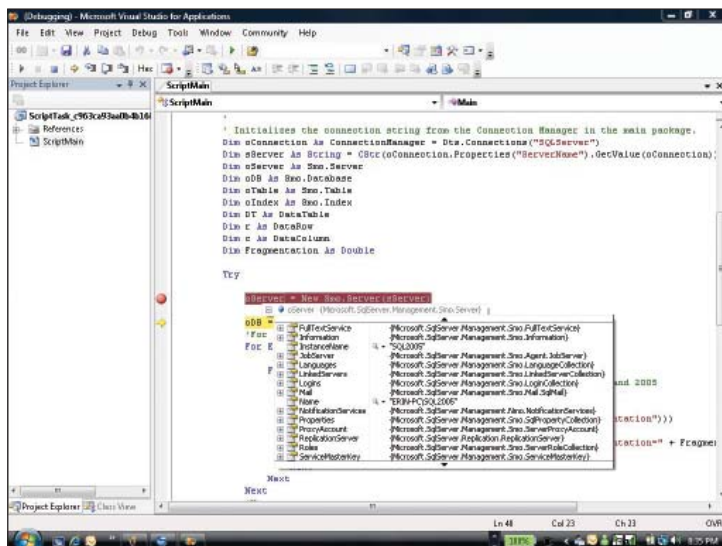


Figure 3

Revealing an object's properties in a breakpoint session

pauses, and the Debug windows give you access to additional information such as variable values, call stacks, and status messages. You can set breakpoints at the package, container, task, or Script task level. You can even set breakpoints to pause after a given number of encounters with an event, such as on the tenth iteration of a task in a For Loop construct. Figure 3 shows how you can interrogate an object to reveal its property values during a breakpoint session.

I find the Locals window, which is accessible from the Debug/Windows/Locals menu item, to be the most useful of SSIS's Debug windows. It displays values for all the variables in the package and even lets you modify variable values, which can be useful when skipping ahead to a later point in the package or simulating certain conditions during package development.

## Debugging Data Flow Tasks

Data Flow tasks control how and when data is manipulated in the package. The primary debugging tool available in Data Flow tasks is the data viewer. I use the data viewer during the development of Data Flow tasks to see what the data looks like in the pipeline as it flows from one task to another—usually just before the task that actually loads the data into its destination.

You can add a data viewer between two Data Flow tasks by right-clicking the connector between the tasks, selecting Data Viewers, and then selecting the data viewer you want to use. You can view the data in a grid, chart, scatter plot, or histogram. I usually view data in a grid, as shown in Web Figure 1 ([www.sqlmag.com/InstantDoc/ID98167](http://www.sqlmag.com/InstantDoc/ID98167)), but I recommend playing with all four data viewers to get a feel for when you should use each one. Although I typically remove my data viewers before deploying a package in production, you don't have to do so.

You can modify columns displayed by a data viewer either as you set up the data viewer or after setup. To modify a data viewer after you've created it, right-click the connector, select Data Viewers, highlight the data viewer, then click Configure.

Another Data Flow task debugging technique that I use frequently during package development and debugging is the RowCount task. The RowCount task relates only to Data Flow tasks. The RowCount task isn't usually billed as a debugging tool but can be quite useful as one. The RowCount task simply counts the rows passed through the pipeline and puts the final row count into a variable. I almost always use the RowCount task as my initial data destination because it serves as a way of examining the data, via a data viewer, without actually loading the data anywhere. Because the RowCount task carries no measurable overhead, it can also be used for baselining or to diagnose performance problems. For more information about using the RowCount task, see the Microsoft white paper "Integration Services: Performance Tuning Techniques" ([www.microsoft.com/technet/prodtechnol/sql/2005/ssisperf.mspx](http://www.microsoft.com/technet/prodtechnol/sql/2005/ssisperf.mspx)).

## Error Logging

Because you can't foresee all the conditions that will ultimately occur in a production environment, SSIS provides powerful and flexible logging capabilities to display information about a package after it's been executed. The challenge is to log enough information



to help you quickly diagnose and minimize the impact of problems that might occur in production.

Several error log providers let you specify where log messages will be written—to text files, SQL Server Profiler, a SQL Server table, the Windows event log, or XML files. You can indicate for which events log messages will be written and the amount of information that's written. The type of events that trigger log messages can vary based on the Control Flow task type. For instance, a Data Flow task can log events such as `OnPipelineRowsSent` and `PipelineInitialization`.

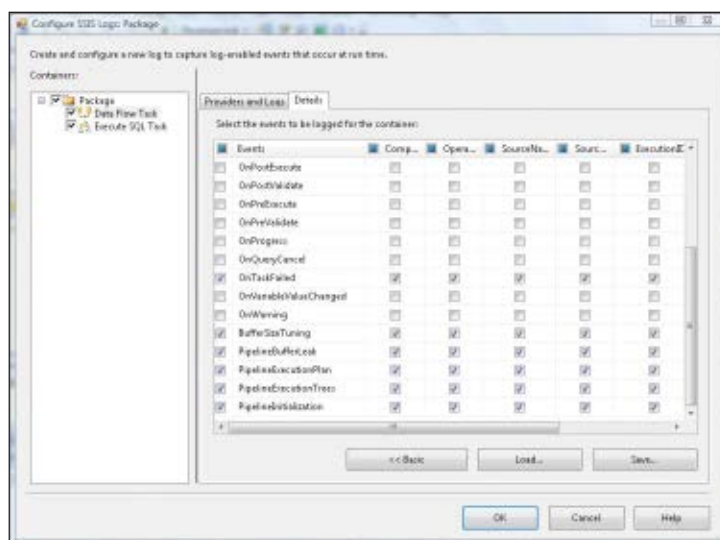
To add logging to a package, click Logging on the SSIS menu and select one or more error log providers (i.e., log entry destinations), which enable you to write to a target destination. Select the check box next to the events you want to log, then click the Details tab and specify the events you want to log. Next, click Advanced to specify the columns to be logged, otherwise all the columns will be logged. After configuring logging, you can view log events in real time during development runs by right-clicking in the Control Flow designer and selecting Log Events.

Because the error log identifies each logged task by name, I recommend implementing a naming standard that uniquely identifies the running SSIS package. Depending on the events you choose to log, the error log can grow fairly rapidly. Be sure to log only the events that you need and occasionally prune old log entries. Because SSIS doesn't include a process to do this out of the box, you must manually prune logs if your error log provider doesn't provide the functionality to do so. For example, if the error log provider is configured to send log messages to SQL Server, rows in the `msdb.dbo.sysdtslog90` table can be deleted after a specified period of time. You can also create logging configurations as templates to provide a consistent strategy across packages that perform similar functions and make log management easier.

## Checkpoints

Checkpoints, another powerful SSIS feature, let you restart a failed package in production. SSIS package execution—especially extraction, transformation, and loading (ETL) package execution—can be lengthy. By its nature, ETL moves and transforms large amounts of data, which can be time-intensive. The failure of a package or task two hours into an ETL process could be catastrophic in a data warehouse system that's required to be available by a set time. Checkpoints let you save the work that's been accomplished so that when you restart the package after resolving the problem that caused it to fail, the process can pick up where it left off. Checkpoints aren't enabled by default, however; they have to be turned on for each package.

Knowing how checkpoints work before you develop your packages is important because they can affect



**Figure 4**

The Details tab showing the error logs to be written

package design. Note that checkpoints can only be applied to Control Flow tasks. For example, a package can't be restarted in the middle of a Data Flow task, which is considered to be a unit of work that's either entirely successful or not. The lack of ability to checkpoint Data Flow tasks provides a good argument for componentizing packages to break logical tasks into Data Flow tasks or, ideally, to try to group packages together based on a Data Flow tasks' packages. For example, you might modularize the packages by taking one big package and making into several smaller packages grouped together inside a controlling package. Then you could set a checkpoint on each module package.

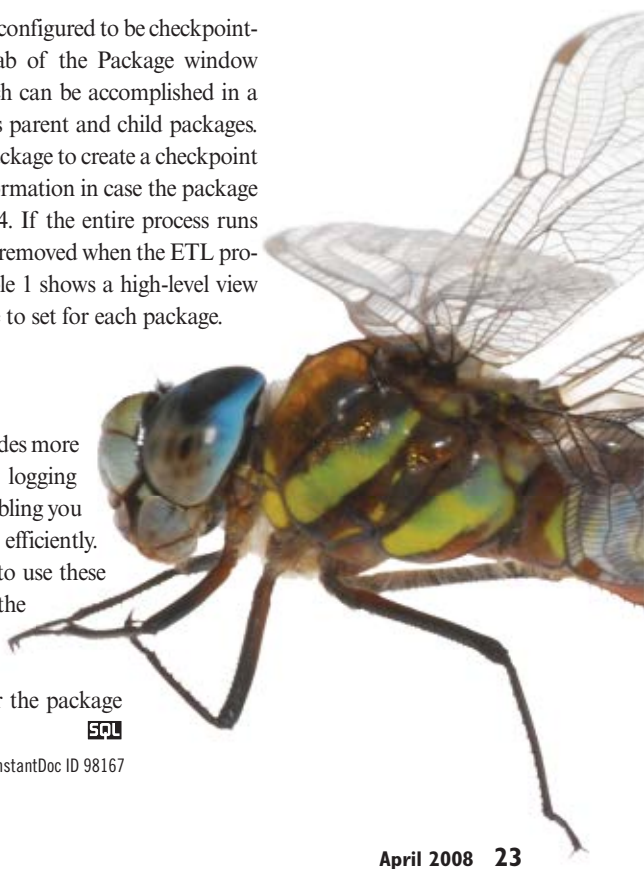
Each package must be configured to be checkpoint-capable on the Details tab of the Package window (shown in Figure 4), which can be accomplished in a SSIS process that includes parent and child packages. You can configure each package to create a checkpoint log to track execution information in case the package fails, as shown in Figure 4. If the entire process runs without error, the logs are removed when the ETL process is complete. Web Table 1 shows a high-level view of the properties you have to set for each package.

## Debugging and Logging in SSIS

As you can see, SSIS provides more powerful debugging and logging capabilities than DTS, enabling you to debug packages more efficiently. Knowing when and how to use these tools can greatly reduce the time it takes to develop packages and resolve problems before and after the package is sent to production.

SQL

InstantDoc ID 98167



**YOU'VE NEVER SEEN A REPORT DO THIS BEFORE.**



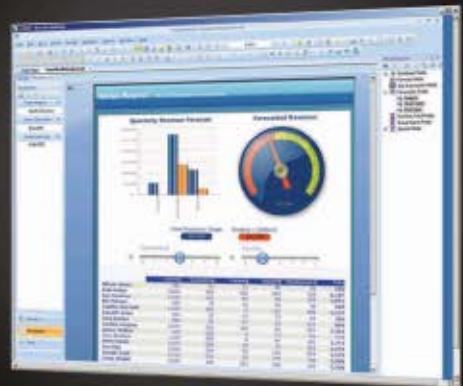
## **CRYSTAL REPORTS® 2008**

### **DEFY THE LAWS OF REPORTING.**

Add new levels of decision support, stunning visualization and rich interactivity to your applications. Discover the latest designer productivity features, an improved report viewing experience and a free runtime for unlimited internal report engine deployment.

#### **Add Crystal Reports to your development tool kit and...**

- Enable What-If analysis with Xcelsius components, right on your reports (as shown).
- Guide report exploration with on-report sorting, filtering and reformatting without re-hitting your database.
- Embed Flash files for stunning visualizations and powerful decision support.



**Explore the new laws of reporting from Crystal Reports.**

**Visit [businessobjects.com/CR2008/dev](http://businessobjects.com/CR2008/dev) or contact us at 1-888-333-6007.**

# Dissecting SQL Server's Top Waits

## What are you waiting for?

**O**kay. You've read "Getting to Know Wait Stats" (October 2007, InstantDoc ID 96746). You know about wait stats—what they are, how to collect them, and how to report on them. Now that you have that foundational knowledge, I want to concentrate on a few wait stats that typically account for most overall system waits. While doing recent performance and scalability audits for clients, I noticed a common pattern regarding the most frequent waits I'm seeing. So, I felt the time was ripe to talk about each.

Let's start the discussion by taking a look at Table 1, which displays a list of 14 distinct wait types, in descending order of total wait time. Of course, this table is merely a partial list of the total wait types; your top waits might or might not be the same as these. However, it's likely that many of these wait types are indeed part of your top waits. For the purposes of this article, let's group these 14 wait types into five unique categories: latches, locking, network, I/O, and parallelism.

### Latches

The LATCH\_xx and PAGELATCH\_xx waits aren't related to physical I/O activities. LATCH\_xx waits are typical of contention for internal resources or structures other than the buffer pool, particularly when you're using heaps or text datatypes. Addressing the problems associated with the other wait types will generally also help pure LATCH waits. PAGELATCH\_xx waits are directly related to contention in the memory caches, including the buffer pool. High waits can indicate one of several problems. One potential problem is inadequate amounts or poorly optimized memory, in which there's a lot of contention for the existing pages in the caches. Adding more memory or tuning your existing system usually helps. Another cause might be extremely high volumes of inserts in the same range of pages, particularly if it results in page splits. This problem can be further complicated if there are many selects on the same range. (You would need to have many thousands of transactions per second to see this problem occur in SQL Server 2005.) If this is the problem, some index tuning might be necessary. PAGELATCH\_UP is an allocation type of wait and typically shows up in the tempdb database under heavy use. Adding more data files to tempdb can significantly reduce or eliminate these waits.

### Locking

The LCK\_xx waits are associated with locking and blocking. High volumes of these waits indicate that something is preventing the efficient execution of the storage engine's locking mechanisms. This problem is mostly caused by transactions that are too long or by a lack of proper index usage, resulting in locking or reading more rows than necessary, thus prolonging the transaction and increasing the chances of further blocking. Poorly configured hardware can also contribute to high LCK\_xx waits because each action simply takes longer to finish, often leading to blocking on multiuser systems.

### Network

Most people assume that high network waits are the result of problems directly related to the physical network between the server and the clients. Although this problem can affect the number of NETWORKIO waits, the reality is that high network waits are most likely caused by poor client response. If the client can't process the results as fast as SQL Server can send them, you'll start to see these waits. Poorly written client



**Andrew J. Kelly**

(AKelly@SolidQ.com) is a SQL Server MVP and the practice manager for performance and scalability at Solid Quality Mentors. He has 20 years experience with relational databases and application development. He is a regular speaker at conferences and user groups.

**TABLE 1: Long System Waits**

Wait Type	Requests	Wait Time	Category
CXPACKET	109,896,512	4,103,400,192.0	Parallelism
PAGEIOLATCH_SH	250,173,264	2,370,188,288.0	I/O
LATCH_EX	832,524,864	878,224,896.0	Latches
NETWORKIO	20,665,884	436,600,832.0	Network
LCK_M_IS	12,380	363,014,656.0	Locking
IO_COMPLETION	43,568,792	265,779,440.0	I/O
LCK_M_S	5,713	228,289,552.0	Locking
LCK_M_X	18,544	191,579,792.0	Locking
PAGEIOLATCH_EX	10,544,062	126,651,336.0	I/O
WRITELOG	22,744,336	114,904,816.0	I/O
LCK_M_SCH_S	884	86,438,568.0	Locking
LCK_M_IX	6,342	70,003,264.0	Locking
PAGELATCH_UP	2,371,233	1,766,169.0	Latches
PAGELATCH_SH	6,534,767	1,563,066.0	Latches



applications and overburdened hardware on the client side can lead to high network waits.

## I/O

The PAGEIOLATCH\_xx, IO\_COMPLETION, and WRITELOG waits fall into the I/O category. PAGEIOLATCH\_xx waits are specific to disk-to-memory transfers and almost certainly indicate problems with your disk subsystem. At the very least, they suggest that your subsystem is having trouble keeping up with the volume of I/O requests. IO\_COMPLETION simply means that the system is waiting for I/O tasks to finish. Finally, WRITELOG indicates that the system is waiting for transaction log-related write requests to finish. You can reduce most of these waits by using one of several tactics:

- Tune your queries so that they read less data to satisfy the requests
- Change the placement of the data and log files so that they don't compete with each other on any particular drive array (e.g., separate the files onto separate physical drives)
- Change the RAID array types and configurations to improve performance (e.g., add more disks, change the array from RAID 5 to RAID 10)

## Parallelism

CXPacket waits are a direct result of parallel processing—more specifically, *inefficient* parallel processing. This wait appears when one thread in a parallel plan is waiting on one or more other threads to finish before it can proceed. High totals of this wait are sure-fire indications that you might not be benefiting much from the use of multiple processors or many multiple processors in parallel. Consider lowering the MAXDOP setting either at the server level or by using query hints (if you can narrow down the culprits to particular queries). See “Boost Performance with Parallel Processing” (InstantDoc ID 97044) for more details.

## Further Study

The Microsoft article “Description of the waittype and lastwaittype columns in the master.dbo.sysprocesses table in SQL Server 2000 and SQL Server 2005” (support.microsoft.com/kb/822101) goes into much more detail about the different waits. And, as always, be sure to visit our Performance Tuning and Optimization forum (sqlforums.windowsitpro.com/web/forum) to discuss this topic further.



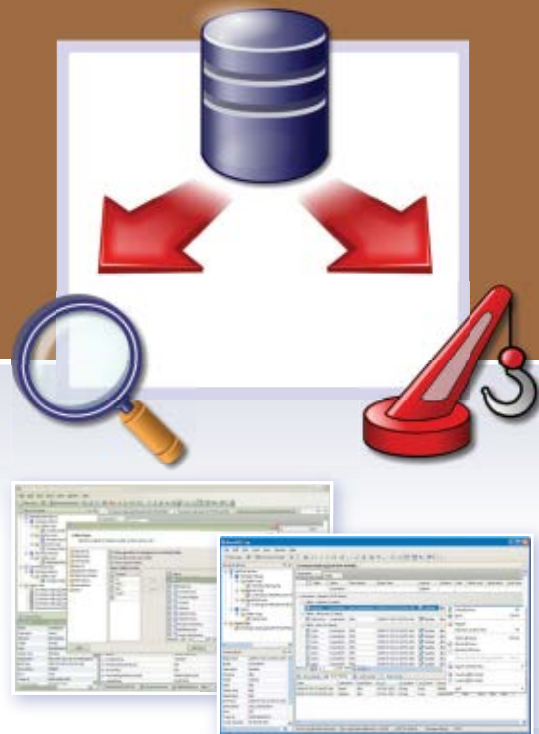
InstantDoc ID 98112



## Audit Data changes, Recover without a backup

### ApexSQL Log the Ultimate Log Reading, Auditing and Recovery tool for SQL Server

- ✓ Recover truncated, deleted or modified data
- ✓ Recover data from corrupted database files
- ✓ Recover Dropped tables
- ✓ Analyze row history for historical changes
- ✓ Generate UNDO and REDO scripts
- ✓ Export transactions to XML, BULK SQL etc
- ✓ Selectively recover data without a backup.



**ApexSQL**  
software

[www.apexsql.com](http://www.apexsql.com)  
or phone 866-665-5500

# Operations

## SQL Server 2005 tips

**S**et operations are fundamental operations in set theory that let you unify sets, figure out the intersection of sets, and determine which elements appear in one set but not the other. SQL Server 2005 supports three set operations, called UNION, INTERSECT, and EXCEPT. A set operation operates on the result sets of two queries (call them query1 and query2), comparing complete rows. The general form of a set operation is

```
query1
set operation: {UNION | INTERSECT |
EXCEPT}
query2
[ORDER BY ...]
```

The set operation you use and the result of the row comparisons determines whether a source row appears in the result of the set operation. UNION will return a row if it appears in either input set, INTERSECT will return a row if it appears in both input sets, and EXCEPT will return a row if it appears in the first set but not the second.

In this article I provide a few set operation tips that you might not be aware of, including circumventing unsupported logical phases, INTERSECT ALL and EXCEPT ALL, and treatment of NULLs. I assume you're already familiar with the fundamentals of set operations; for more information about set operations, see the Learning Path, page 28.

### Circumventing Unsupported Logical Phases

As you can see from the general form of a set operation that I provided, only one logical query processing phase is allowed on the result of a set operation—ORDER BY. The syntax of a set operation doesn't allow applying any other logical phases to the result (e.g., table operators, WHERE, GROUP BY, HAVING, TOP). However, you can use a simple trick

to get around this restriction—define a table expression (e.g., common table expression—CTE—or derived table) based on the query with the set operation, then have the outer query against the table expression apply any logical phases/query clauses that you need. For example, if you run the query that Web Listing 1 shows ([www.sqlmag.com](http://www.sqlmag.com), InstantDoc ID 98159) in the Northwind database, it returns customer-employee pairs that had order activity in both 1996 and 1997.

Suppose you need to group the result by employee, and count the number of customers each employee handled. This can be achieved easily with a table expression such as the one that Web Listing 2 shows. In a similar manner, you can apply any other logical phase to the table expression C.

Going back to the general form of the set operation, query1 and query2 aren't allowed to have ORDER BY clauses because by definition a set operation operates on unordered sets, and an ORDER BY clause would impose order. This restriction applies even when you use the TOP option. Suppose you need to apply a set operation to the result sets of two queries defined with a TOP option and an ORDER BY clause. To accomplish this task, you must define two table expressions based on the TOP queries, and have the set operation applied to two queries against the table expression, as Web Listing 3 shows. This query returns shipping destinations (country, region, city) that were in the top five shipping destinations in 1997 (based on number of orders) but weren't in the top five in 1996.

### INTERSECT ALL and EXCEPT ALL

ANSI SQL supports two versions of each set operation—a DISTINCT version and an ALL version. SQL Server 2005 implements both versions of the UNION set operation, but only the DISTINCT version of INTERSECT and EXCEPT. In this section I explain



**Itzik Ben-Gan**

([itzik@solidqualitylearning.com](mailto:itzik@solidqualitylearning.com)), a mentor at Solid Quality Learning, teaches, lectures, and consults internationally. He manages the Israeli SQL Server Users Group, is a SQL Server MVP, and is the author of the *Inside Microsoft SQL Server 2005: T-SQL* series (MSPress, 2006).



### MORE on the WEB

Download the Web listings and see the Web tables at InstantDoc ID 98159.

what the two versions mean with each set operation, and I present alternatives you can use for the missing ALL versions.

The DISTINCT version of INTERSECT means that as long as a row appears at least once in each set, the row will appear once in the output. For example, the following query returns one occurrence of each location (country, region, city) that has at least one customer and one employee:

```
SELECT Country, Region, City FROM
    dbo.Customers
INTERSECT
SELECT Country, Region, City FROM
    dbo.Employees;
```

This query returns three rows in the output, where each qualifying location appears only once.

ANSI SQL also supports an INTERSECT ALL set operation that wasn't implemented in SQL Server 2005. INTERSECT ALL returns as many occurrences of a row as the minimum number of times that it appears in any set. If row *R* appears *m* times in the result set of query1, and *n* times in the result set of query2, *R* will appear a minimum number of times (*m*, *n*) in the result of the set operation. For example, if *R* appears three times in one set and five times in the other, *R* will appear three times in the result of the set operation. You typically need to use INTERSECT ALL if you want to pair each instance of a row from one set with a different instance of the row in the other. Using our last query as an example, say that each employee is supposed to handle only one customer from the same location, and you need to check how many customers can be matched with an employee from the same location. In this case, you'd need INTERSECT ALL, as in the following code (don't run this query):

```
SELECT Country, Region, City FROM
    dbo.Customers
INTERSECT ALL
SELECT Country, Region, City FROM
    dbo.Employees;
```

Unfortunately, SQL Server 2005 doesn't support this standard INTERSECT ALL syntax. A simple alternative is to assign row numbers to number the occurrences of each location, and issue the INTERSECT

operation between the two sets that include the row numbers, as Web Listing 4 shows.

As an example, the location UK, NULL, London appears six times in Customers and four times in Employees. The query against Customers assigns row numbers 1 through 6 to the occurrences of this location, and the query against Employees assigns row numbers 1 through 4 to the occurrences of this location. Four occurrences of this location (with row numbers 1 through 4) intersect; therefore this location will appear four times in the output.

Similarly, ANSI SQL defines a set operation called EXCEPT ALL that isn't implemented in SQL Server 2005. If row *R* appears *m* times in the result of query1 and *n* times in the result of query2, *R* will appear a maximum number of times (*m-n*, 0) in the result of query1 EXCEPT ALL query2. For example, a location *L* that appears six times in the result of query1 and four times in the result of query2 will appear two times in the result of query1 EXCEPT ALL query2, and zero times in the result of query2 EXCEPT ALL query1.

You typically need EXCEPT ALL when you want to figure out how many instances of a row from one set can't be paired with a different instance of the row in the other set. Using our last query as an example, say that each employee is supposed to handle only one customer from the same location, and you need to check how many customers can't be matched with an employee from the same location.

To achieve the logical equivalent of the missing EXCEPT ALL operation, you can use the same trick with the row numbers I showed earlier, only this time you use the EXCEPT operation instead of INTERSECT, as Web Listing 5 shows.

As I mentioned earlier, the location UK, NULL, London appears six times in Customers and four times in Employees. The occurrences of this location are numbered 1 through 6 in the first set and 1 through 4 in the second. Only occurrences 5 and 6 from the first set don't find a match in the second; therefore in the 85 output rows you'll find only two occurrences of UK, NULL, London.

## Treatment of NULLs

One of the key benefits of using set operations over alternative methods is that a set operation treats NULLs as equal when comparing rows. Take the following query as an example:

```
SELECT Country, Region, City FROM
    dbo.Customers
INTERSECT
SELECT Country, Region, City FROM
    dbo.Employees;
```

This query returns distinct locations where both employees and customers exist. Web Table 1 shows



## LEARNING PATH

### SQL SERVER MAGAZINE RESOURCES

"Don't Avoid the UNKNOWN," InstantDoc ID 47010

"Get in the Loop with CTEs," InstantDoc ID 42072

"GROUP BY ALL," InstantDoc ID 50500

"Set-Operation Alternatives," InstantDoc ID 40321




the output of the query. Notice that the location UK, NULL, London appears in the output because both customers and employees exist in this location. The set operation considers the NULL region in the customer location (UK, NULL, London) as being equal to the NULL region in the employee location (UK, NULL, London).

Other language elements such as ON and WHERE filters have a different treatment when comparing two NULLs. For example, an attempt to write a solution that uses an INNER JOIN statement, as Web Listing 6 shows, would return the output shown in Web Table 2. Notice that the location UK, NULL, London doesn't appear in the output even though it does appear in both Customers and Employees. This location was eliminated by the INNER JOIN statement. When the JOIN predicate in the ON clause compared the NULL region in both sides, the comparison yielded UNKNOWN. An ON filter (as well as WHERE and HAVING) eliminates rows for which the predicate yields UNKNOWN. To get the exact logical equivalent of the INTERSECT query, you'd need to explicitly handle NULLs as in Web Listing 7. At this point, the solution is so convoluted that you can really appreciate the brevity and simplicity of the INTERSECT set operation.

In a similar manner, the EXCEPT operation will also treat two NULLs as equal. For example, the following query returns the output that Web Table 3 shows, with the distinct locations where employees exist but not customers:

```
SELECT Country, Region, City FROM
    dbo.Employees
EXCEPT
SELECT Country, Region, City FROM
    dbo.Customers;
```

Notice that the location UK, NULL, London doesn't appear in the output because it *does* appear in both Employees and Customers. Now run the alternative solution in Web Listing 8, using the NOT EXISTS predicate. You'll get the output that Web Table 4 shows.

This time the location UK, NULL, London was returned even though it appears in both Employees and Customers. When the inner query's filter compared the NULL region in the customer row and the NULL region in the employee row, the predicate evaluated to UNKNOWN, and the row was filtered out as if London, NULL, UK doesn't exist in the Customers table. Hence, the location was returned from the Employees table. To get the true logical equivalent of the EXCEPT query, you'd need to handle NULLs explicitly, as in Web Listing 9. But again, this alternative solution is quite convoluted, emphasizing the advantage of using the shorter, simpler, and more elegant EXCEPT version. 

InstantDoc ID 98159



## THE LOGICAL PUZZLE

### Solution to March's Puzzle: Too Clever by Half

A chicken and a half lay an egg and a half in a day and a half. How many eggs would one chicken lay in three days? A builder and a half build a house and a half in a year and a half using a tool and a half. How many houses would one builder build in nine years? Can you generalize your calculation to solve both equations?

The intuitive yet incorrect answer to the chicken and eggs puzzle is three eggs. Our brain plays a trick on us and makes us think that if a chicken and a half lay an egg and a half in a day and a half, one chicken lays one egg in one day. But if you express the relationship between chickens, days, and eggs mathematically, you get the equation

$$3/2 \text{ chickens} \times 3/2 \text{ days} = 3/2 \text{ eggs}$$

Reducing the number of chickens from 3/2 to 1 is achieved by dividing the original number by 3/2. For the equation to be true, you also need to divide the number of eggs (3/2) by 3/2, giving you the equation

$$1 \text{ chicken} \times 3/2 \text{ days} = 1 \text{ egg}$$

Reducing the number of days from 3/2 to 1 has a similar effect on the number of eggs; namely, you need to divide 1 (egg) by 3/2, giving you the equation

$$1 \text{ chicken} \times 1 \text{ day} = 2/3 \text{ egg}$$

If you increase the number of days from 1 to 3, the effect on the number of eggs is a factor of 3 as well:

$$1 \text{ chicken} \times 3 \text{ days} = 2 \text{ eggs}$$

In a very similar manner you can express the relationships between builders, houses, years, and tools with the following equation:

$$3/2 \text{ builders} \times 3/2 \text{ years} \times 3/2 \text{ tools} = 3/2 \text{ house}$$

To reduce the number of builders, years, and tools to one each, you need to divide the number of houses by 3/2 three times; in other words, by (3/2)^3:

$$1 \text{ builder} \times 1 \text{ year} \times 1 \text{ tool} = 3/2 \div 3/2 \div 3/2 \div 3/2 \text{ houses}$$

Giving you the equation

$$1 \text{ builder} \times 1 \text{ year} \times 1 \text{ tool} = 4/9 \text{ house}$$

Thus, one builder with one tool will build four houses in nine years.

To generalize the equation, you need to divide the right side of the equation by 3/2 *n* times for *n* elements in the left side of the equation. Or, if you want to express the calculation as a multiplication instead of division, multiply by (2/3)^*n*. For example, in our last equation:

$$3/2 \text{ builders} \times 3/2 \text{ years} \times 3/2 \text{ tools} = 3/2 \text{ houses}$$

The left side of the equation contains three elements; therefore

$$1 \text{ builder} \times 1 \text{ year} \times 1 \text{ tool} = 3/2 \times (2/3)^3 \text{ houses}$$

which is equal to

$$1 \text{ builder} \times 1 \text{ year} \times 1 \text{ tool} = 4/9 \text{ house}$$

### April's Puzzle: Suicidal Mosquito

Two trains drive toward each other on the same rail. Both trains drive at a speed of 100MPH. When the trains are 100 miles apart, a mosquito starts flying back and forth from the front of one train to the other at a speed of 200MPH. What total distance will the mosquito cover before the two trains crash?

InstantDoc ID 98160



# No servers run cooler.

Fujitsu PRIMERGY® servers beat the heat with Cool-Safe™ technology.

IT organizations have always counted on the performance and reliability of Fujitsu PRIMERGY servers. Thanks to Cool-Safe technology, they can count on lower utility bills and a smaller carbon footprint, too. Even cooler, our new PRIMERGY TX120 tower server for small businesses and SOHOs is the smallest, quietest, most energy-efficient tower server on earth. And you don't even need an on-site IT staff because Fujitsu PRIMERGY servers include PRIMERGY Server View Remote Management software, which gives outside consultants or IT staff at headquarters all the secure remote access they need. How cool is that? To find out, go to <http://us.fujitsu.com/computers/PRIMERGY>.

#### HIGH ENERGY-EFFICIENCY.

A Fujitsu PRIMERGY TX120 tower server with a Dual-Core Intel® Xeon® UP processor uses up to 40 percent less power than previous towers.

#### COOL-SAFE TECHNOLOGY.

Bigger fans and independent air tunnels improve airflow, so Fujitsu PRIMERGY servers can run faster, cooler, and quieter.



BX620 S4 Blade Servers



TX120  
Tower Server



TX300 S3  
Tower Server

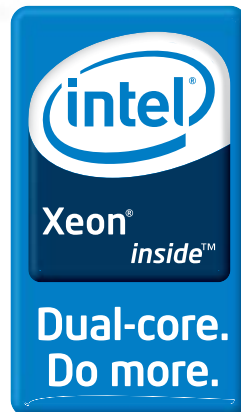


RX300 S3 Rack Server

**FUJITSU**

THE POSSIBILITIES ARE INFINITE

© 2008 Fujitsu Computer Systems Corporation. All rights reserved. Fujitsu and the Fujitsu logo are registered trademarks of Fujitsu Limited. Cool-Safe is a trademark, and PRIMERGY is a registered trademark, of Fujitsu-Siemens Computers GmbH. Intel, Intel Logo, Intel Inside, Intel Inside Logo, and Xeon are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries. All other trademarks and product names are the property of their respective owners.



# T-SQL 101, Lesson 2

## How to use the INSERT, UPDATE, and DELETE statements

**An** important part of many DBAs' and SQL Server administrators' jobs is keeping databases current. Doing so involves adding, changing, and deleting data from tables. T-SQL makes these tasks easy. You just need to use INSERT, UPDATE, and DELETE statements. Before I show you how to use these statements, there are a few things you need to do before proceeding.

### The Prerequisites

The sample INSERT, UPDATE, and DELETE statements I use in this article are run against a simple database named MyDB, which contains a few dummy tables and a table named MyTable. MyTable is used to store the names of the various dummy tables. The only purpose of the dummy tables is to take up space. By merely existing, they ensure that you'll have data in the sysobjects table that you can use as a data source for inserting data into MyTable.

Assuming that you have the permissions needed to create databases and tables and to issue data-modification statements, you can create MyDB, MyTable, and the dummy tables with the MyDB.sql and MyTable.sql files. You can download these files by going to [www.sqlmag.com](http://www.sqlmag.com), entering 98105 in the InstantDoc ID text box, and clicking the 98105.zip hotlink. After you've downloaded and unzipped the 98105.zip file, open a query window in either SQL Server 2005's SQL Server Management Studio (SSMS) or SQL Server 2000's Query Analyzer and copy the code in MyDB.sql into the window. In the two FILENAME entries, replace *C:\Program Files\Microsoft SQL Server\MSSQL\1\MSSQL\DATA\*

*MyDB.mdf* with a valid path on your server. Run the query to create MyDB. To create MyTable and the dummy tables, copy the code in MyTable.sql into the query window and execute it.

Before working with data in any table, you should familiarize yourself with the table's layout. So, open MyTable and do so.

### Inserting a Single Record

A basic INSERT statement that adds a single record (aka row) to a table consists of three parts and looks like

```
INSERT INTO Part1
      (Part2)
VALUES (Part3)
```

Part1 is where you specify the target table that will hold the new record. Part2 is where you list the names of the columns in the target table for which you have data. You need to enclose the list in parentheses and use commas to separate column names. Part3 is where you provide the data to be inserted.

When you want to specify the actual data values, you use the VALUES parameter followed by those values.

You need to enclose the list of data values in parentheses and separate the values with commas. When a value is character based—such as a string or a datetime value—you also need to enclose it in single quotes (''). For example, running the following statement inserts the data values of TestTable and 2007-09-22 into MyTable's ObjectName and CreationDate columns, respectively:

```
INSERT INTO MyTable
      (ObjectName, CreationDate)
VALUES ('TestTable', '2007-09-22')
```

If you familiarized yourself with MyTable earlier, you probably noticed that I only specified two of the



**Bill McEvoy**

([bill@cookingwithsql.com](mailto:bill@cookingwithsql.com)) is the Master Chef/DBA for Cooking With SQL. Having been a DBA since SQL Server 4.2, he specializes in batch processing and performance tuning.

### MORE on the WEB

Download the code at  
InstantDoc ID 98105.



### LEARNING PATH

#### SQL SERVER MAGAZINE RESOURCES

"T-SQL 101, Lesson 1," InstantDoc ID 97724



three columns in this INSERT statement. The first column is an identity column, which has a special meaning. The data for that column is automatically added and its values incremented with each record being inserted. This is a powerful feature of SQL Server that you should be aware of. If you want to learn more about identity data types, take a look at the *IDENTITY* (Property) Web page in SQL Server Books Online (BOL) at [msdn2.microsoft.com/en-us/library/ms186775.aspx](http://msdn2.microsoft.com/en-us/library/ms186775.aspx).

To view the newly inserted record, execute the code

```
SELECT * FROM MyTable
```

Figure 1 shows what the results should look like.

TableID	ObjectName	CreationDate
1	TestTable	2007-09-22 00:00:00.000

Figure 1

Results from using the INSERT statement to add a single record

## Inserting the Results from a SELECT Statement

In “T-SQL 101, Lesson 1” (March 2008, InstantDoc ID 97724), I showed you how to get data out of a database with a SELECT statement. You can use the INSERT statement to store the data retrieved by the SELECT statement in a table. Like the single-record INSERT statement, the INSERT statement that stores the results of a SELECT statement has three parts: the target database (Part1), the column names (Part2), and the data (Part3). However, instead of using the VALUES parameter to specify the actual data values in Part3, you use a SELECT statement that retrieves the data from another source. For example, when run against the MyDB database, the query

```
INSERT INTO MyTable
(ObjectName, CreationDate)
SELECT name, crdate
FROM sysobjects
WHERE type = 'U'
ORDER BY name
```

**Use your newfound knowledge with caution. You don't want to be inserting millions of records at a time on a server that is busy or has limited disk space.**

inserts into our new table MyTable records that contain the name and creation date of all the user-defined tables in MyDB, ordered alphabetically by their names. So, assuming you ran the single-record INSERT statement previously discussed, your results from this query should look like those in Figure 2, with two exceptions. First, the CreationDate values for the MyTable and

TableID	ObjectName	CreationDate
1	TestTable	2007-09-22 00:00:00.000
2	EventLog	2007-10-08 23:32:34.500
3	MyProcedure	2007-10-08 23:32:34.500
4	MyTable	2007-10-08 23:32:34.500
5	MyTrigger	2007-10-08 23:32:34.500
6	MyView	2007-10-08 23:32:34.500

Figure 2

Results from using the INSERT statement to store the data retrieved by a SELECT statement

dummy tables will contain the date and time when you ran MyTable.sql. Second, the ObjectName column will be wider. (I shortened it for space reasons.)

In Figure 2, note that TestTable is still the first record. The ORDER BY name clause only applies to the new records that the SELECT statement is inserting. TestTable was added previously with the single-record INSERT statement.

You can use any SELECT statement you like, as long as the data types of the columns listed in Part2 of the INSERT statement match those columns specified in the SELECT statement. This opens the door for capturing all sorts of data. Use your newfound knowledge with caution, however. You don't want to be inserting millions of records at a time on a busy server or on a server with limited disk space.

## Updating Data

Now that you have a handle on inserting data, let's look at how the UPDATE statement works. A simple UPDATE statement typically consists of three parts:

```
UPDATE Part1
SET Part2
WHERE Part3
```

Part1 is where you specify the target table. Part2 is where you specify the columns that are to be changed, along with the new data for each column. Part3 is optional but in most cases essential. This is where you specify a filter using the WHERE clause. If you don't specify a WHERE clause, you'll update every single record in the table. For example, the query

```
UPDATE MyTable
SET CreationDate = '2007-09-23'
```

TableID	ObjectName	CreationDate
1	TestTable	2007-09-23 00:00:00.000
2	EventLog	2007-09-23 00:00:00.000
3	MyProcedure	2007-09-23 00:00:00.000
4	MyTable	2007-09-23 00:00:00.000
5	MyTrigger	2007-09-23 00:00:00.000
6	MyView	2007-09-23 00:00:00.000

Figure 3

Results from updating the CreationDate columns for all the records

updates the `CreationDate` value for each record in `MyTable`, as Figure 3 shows. In Figure 3, note that the `CreationDate` value is 2007-09-23 00:00:00.000 and not 2007-09-23 as specified in the query. Because the data type of the `CreationDate` column is defined as `datetime` and the query doesn't specify a time portion, SQL Server assumes you mean midnight and adds 00:00:00.000.

Now let's use a `WHERE` clause to modify the `ObjectName` and `CreationDate` columns for a specific record:

```
UPDATE MyTable
SET ObjectName = 'PartyTime',
    CreationDate = '1999-12-31
    23:00:00'
WHERE TableID = 1
```

As Figure 4 shows, only the first record (specified by `WHERE TableID = 1`) is updated with the new table name of `PartyTime` and a new creation date of 1999-12-31 23:00:00.

## Deleting Data

The command that every DBA fears will fall into the wrong hands is the `DELETE` statement. Although it can be misused, it serves an important role when using queries to modify data in tables. A basic `DELETE` statement typically consists of two parts:

```
DELETE Part1
WHERE Part2
```

`Part1` is where you specify the target table. `Part2` is where you specify a filter using the `WHERE` clause. Like the `WHERE` clause in an `UPDATE` statement, the `WHERE` clause in a `DELETE` statement is optional but usually essential. If you don't include a `WHERE` clause and filter, you'll delete *all* the records in the specified table.

TableID	ObjectName	CreationDate
1	PartyTime	1999-12-31 23:00:00.000
2	EventLog	2007-09-23 00:00:00.000
3	MyProcedure	2007-09-23 00:00:00.000
4	MyTable	2007-09-23 00:00:00.000
5	MyTrigger	2007-09-23 00:00:00.000
6	MyView	2007-09-23 00:00:00.000

**Figure 4**

Results from using a `WHERE` clause to update the `ObjectName` and `CreationDate` columns for a specific record

TableID	ObjectName	CreationDate
1	PartyTime	1999-12-31 23:00:00.000

**Figure 5**

Results from deleting all records that have a creation date older than September 22, 2007

Let's say you want to delete the records with a creation date older than September 22, 2007 from `MyTables`. The query

```
DELETE MyTable
WHERE CreationDate > '2007-09-22'
```

will do just that, as Figure 5 shows.

You need to be careful when modifying data with either the `DELETE` or `UPDATE` statement. Always include a `WHERE` clause unless you're positive you want to affect all the records in a table. If you're unsure of how many records you'll be affecting, try running a `SELECT` statement with the same `WHERE` clause. Instead of specifying columns in the `SELECT` statement, you can specify `COUNT(*)`, which will return a count of the rows that would be affected. For example, to check the effect of the last `DELETE` statement, you could run

```
SELECT COUNT (*) FROM MyTable
WHERE CreationDate > '2007-09-22'
```

**To learn how to better use the `INSERT`, `UPDATE`, and `DELETE` statements, try running them against a table in one of your existing databases on a nonproduction server. You might even find data that you didn't know existed.**

If the result set is in the millions, chances are you need to refine your `WHERE` clause. I'll cover the use of the `COUNT` function in the next lesson, so don't fret if you can't get it to work.

## Explore on Your Own

Once you're comfortable in modifying the data in `MyTable`, you might try running `INSERT`, `UPDATE`, and `DELETE` statements against tables in an existing database on a nonproduction server. To do so, copy the code in `ExistingDatabaseQuery.sql` (which is in the 98105.zip file) in your query window, replace `MyDB` in the first line to the name of your existing database, and run the query. Afterward, you'll be able to customize the sample `INSERT`, `UPDATE`, and `DELETE` statements to your environment and run them. By doing so, you'll learn how to better use these statements. You might even find data that you didn't know existed.

**SQL**

InstantDoc ID 98105

# Further EXPLORATIONS in Plan-Cache Metadata

Drill deeper into query plan details by using 3 dynamic management views



**Kalen Delaney**

(kalen@insidesqlserver.com) has been working with SQL Server for 20 years and provides SQL Server training and consulting to clients around the world. Her most recent book is *Inside Microsoft SQL Server 2005: Query Tuning and Optimization* (Microsoft Press).

**SQL** Server 2005 provides metadata that gives you information about cached query plans in XML format, so that you can view plans without having to run them yourself or compare multiple plans at once. “Examining XML Plans in Cache,” January 2008, InstantDoc ID 97562 introduced you to working with plan-cache metadata using the `sys.dm_exec_cached_plans` dynamic management view. We’ll continue the discussion by looking at several additional metadata objects that you can use to obtain further information about query plans in your plan cache.

## More Metadata

“Examining XML Plans in Cache” presented a simple query that used the new SQL Server 2005 CROSS APPLY operator to combine `sys.dm_exec_cached_plans` with two functions, `sys.dm_exec_query_plan` and `sys.dm_exec_sql_text`, which Microsoft introduced in SQL Server 2005 SP2. Combining these three objects lets you see reuse information about every cached plan as well as the text of the query and the XML form of the query plan.

The `sys.dm_exec_query_plan` function returns a query plan in an XML-type output column. `sys.dm_exec_text_query_plan` is similar to `sys.dm_exec_query_plan` and can be used in place of `sys.dm_exec_query_plan`. However, the two functions aren’t quite interchangeable because `sys.dm_exec_text_query_plan` requires two additional parameters specifying the start and end positions in the batch where the particular statement you’re interested in can be found. If you use the default values for start and end (actually `statement_start_offset` and `statement_end_offset`), you can get the plan for the entire batch.

There are some other differences between `sys.dm_exec_query_plan` and `sys.dm_exec_text_query_plan`, as described in SQL Server *Books Online* (BOL):

- With `sys.dm_exec_text_query_plan`, the query plan’s

output is returned in text format, rather than XML. Although the returned plan looks like an XML plan, it’s actually contained in a column of type `nvarchar(max)`.

- With `sys.dm_exec_text_query_plan`, the query plan’s output isn’t limited in size, as the XML output of `sys.dm_exec_query_plan` is.
- The `sys.dm_exec_text_query_plan` function lets you specify individual statements within the batch. As I mentioned, this is possible when using non-default values for `statement_start_offset` and `statement_end_offset`, for batches that contain multiple statements. I’ll provide an example using these parameters later in this article.

BOL goes on to say the following about `sys.dm_exec_text_query_plan` (this information isn’t quite correct, though): “When an ad hoc query uses simple or forced parameterization, the `query_plan` column will contain only the statement text and not the actual query plan. To return the query plan, call `sys.dm_exec_text_query_plan` for the plan handle of the prepared parameterized query.”

I discussed the use of such plans in “Examining XML Plans in Cache” and referred to these statement-text-only plans as truncated plans representing what I call shell queries. I also mentioned that plans that have been parameterized aren’t the only ones that will have a truncated plan. Ad hoc plans for unsafe plans will also show only the statement, and in such instances, no prepared plan is available that would enable you to see the entire plan.

## Viewing Autoparameterized Queries in Cache

Let’s look at an example of how `sys.dm_exec_text_query_plan` and `sys.dm_exec_query_plan` display such truncated plans. We’ll run two of the same queries I used in “Examining XML Plans in Cache”—but this



## MORE on the WEB

Download the listings at InstantDoc ID 97975.



time using the `sys.dm_exec_text_query_plan` and `sys.dm_exec_query_plan` functions. Because my batch will have only one statement, I can use the default parameters for the start and end positions in the batch. Run the code in Listing 1 to clear the plan cache and to run two queries that can be autotparameterized to use the same plan. Listing 1 then displays the contents of plan cache and for each plan includes an XML representation using both the `nvarchar` data type and the `xml` data type. Note that the XML data type column is a link that you can click to see the entire XML document.

If no other concurrent activity is occurring on your SQL Server system, you should get three rows returned from the metadata query. The two rows with an `objtype` value of `adhoc` will be the shell queries and contain only the SQL statement itself, not the entire execution plan. Only the row for the prepared query will have the entire plan available in the XML.

When you examine the output of the column called `text_query_plan`, the plan might look incomplete. Examining it in the output from Listing 1 doesn't show the appropriate `</ShowPlanXML>` terminator. However, you should be able to display the entire XML document by copying the single value in the `text_query_plan` column for the Prepared query row and pasting it in another query window.

So where do you get `statement_start` and `statement_end` information if you're interested in only a single statement in a multi-statement batch? Two other metadata objects can help you obtain this information.

### Sys.dm\_exec\_requests

The `sys.dm_exec_requests` dynamic management view returns one row for every currently executing request within your SQL Server instance. `sys.dm_exec_requests` is useful for many purposes besides tracking down plan-cache information. This dynamic management view contains the `sql_handle` and `plan_handle` for the current statement as well as resource-usage information for each request.

For troubleshooting purposes, you can use the `sys.dm_exec_requests` view to help identify long-running queries. This view contains information about each batch that was running and also contains the columns `statement_start_offset` and `statement_end_offset`, which indicate the position within the entire batch of the currently executing statement. The offsets start at 0; an offset of -1 indicates the end of the batch. You can use the statement start and end values as the second and third parameters passed to `sys.dm_exec_text_query_plan` to extract the plan for only the currently executing statements, as the code in Listing 2 shows.

Listing 2 includes a batch that executes three state-

### LISTING 1: Viewing Cached-Plan Metadata for Autoparameterized Queries

```
USE AdventureWorks;
GO
DBCC FREEPROCCACHE;
GO
SELECT * FROM Sales.SalesOrderHeader
WHERE SalesOrderID = 43665;
GO
SELECT * FROM Sales.SalesOrderHeader
WHERE SalesOrderID = 71056;
GO
SELECT plan_handle, usecounts, cacheobjtype,
       objtype, [text],
       tqp.query_plan as text_query_plan,
       xqp.query_plan as xml_query_plan
FROM sys.dm_exec_cached_plans
CROSS APPLY sys.dm_exec_sql_text(plan_handle)
CROSS APPLY sys.dm_exec_text_query_plan
            (plan_handle, 0, -1) tqp
CROSS APPLY
            sys.dm_exec_query_plan (plan_handle) xqp
WHERE cacheobjtype = 'Compiled Plan'
AND [text] NOT LIKE '%sys.dm_%';
```

### LISTING 2: Examining Execution Information for a Statement Within a Batch

```
USE AdventureWorks;
SELECT * FROM Sales.SalesOrderHeader
WHERE SalesOrderID = 71056;
SELECT *
FROM sys.dm_exec_requests
CROSS APPLY
    sys.dm_exec_text_query_plan
        (plan_handle, statement_start_offset,
         statement_end_offset );
```

ments. Your output should contain two result sets. The first will be the output of the `SELECT` from the `Sales.SalesOrderHeader` table, and the second returns a row showing the contents of `sys.dm_exec_requests` for the currently running batch, with the row's last column containing the plan for only the statement that returns the `CROSS APPLY` query. (That is, you won't see the plan for the `SELECT` from `Sales.SalesOrderHeader` because that statement isn't currently running when the `CROSS APPLY` query is run.) Note that including the asterisk (\*) in the `SELECT` list indicates that this query should return all the columns from `sys.dm_exec_requests` view (as well as from `sys.dm_exec_text_query_plan`). You can replace the asterisk with the columns that you're particularly interested in, such as `start_time` or `blocking_session_id`.

You can use `sys.dm_exec_requests` and `sys.dm_exec_text_query_plan` to track down slow-running queries on your production system. Keep in mind that the `sys.dm_exec_sql_text` function doesn't let you specify the start and end offsets, so by default you'll see the SQL text for the entire batch. However, you can use the start and end offsets along with the substring function to manually extract the SQL for the current statement only. The code in Listing 3, page 36, shows you the SQL text and query plan for the three currently executing statements that have the longest

**You can use the `sys.dm_exec_requests` view to help identify long-running queries.**

**LISTING 3: Returning the Top 3 Current Longest-Running Queries**

```

SELECT TOP 3 SUBSTRING(text,
    (statement_start_offset/2) + 1,
    ((CASE statement_end_offset
        WHEN -1
        THEN DATALENGTH(text)
        ELSE statement_end_offset
    END - statement_start_offset)/2) + 1)
    AS query_text, *
FROM sys.dm_exec_requests
CROSS APPLY sys.dm_exec_sql_text
    (sql_handle)
CROSS APPLY
    sys.dm_exec_text_query_plan
    (plan_handle, statement_start_offset,
    statement_end_offset)
ORDER BY total_elapsed_time DESC

```

elapsed times. The SUBSTRING function needs to divide the offset values by two, because SUBSTRING takes the character-position values as parameters, but the values returned in statement\_start\_offset and statement\_end\_offset are the offsets in bytes in string of type nvarchar. Nvarchar requires two bytes per character.

Note that the sys.dm\_exec\_requests view only shows you what's currently running. If you want to see metadata for prior queries, however, you can use another view: sys.dm\_exec\_query\_stats.

**Using sys.dm\_exec\_query\_stats**

For detailed troubleshooting, you can use sys.dm\_exec\_query\_stats to return performance information for individual queries. This view returns performance statistics for cached query plans, aggregated across all executions of the same query. The view contains one row per query statement within each cached plan, and the lifetime of the rows in the view is tied to the plan itself. You'll see information only for plans that are currently still in cache. After a plan is removed from the cache, sys.dm\_exec\_query\_stats will no longer display any information about the plan.

The sys.dm\_exec\_query\_stats view returns both a sql\_handle and a plan\_handle as well as the start and end offsets like those you saw in sys.dm\_exec\_requests. The query in Listing 4 uses sys.dm\_exec\_query\_stats to return the top ten queries by total CPU time. Using

**LISTING 4: Returning the Top 10 Cached Queries with Most CPU Usage**

```

SELECT TOP 10 SUBSTRING(text,
    (statement_start_offset/2) + 1,
    ((CASE statement_end_offset
        WHEN -1
        THEN DATALENGTH(text)
        ELSE statement_end_offset
    END - statement_start_offset)/2) + 1)
    AS query_text, *
FROM sys.dm_exec_query_stats
CROSS APPLY sys.dm_exec_sql_text
    (sql_handle)
CROSS APPLY sys.dm_exec_text_query_plan
    (plan_handle, statement_start_offset,
    statement_end_offset)
ORDER BY total_worker_time/execution_count
DESC;

```

this information, you can identify the most expensive queries that have been run on your SQL Server system and have plans that are still cached.

Sys.dm\_exec\_query\_stats returns a large amount of performance information for each query, including the number of times the query was executed and the cumulative I/O, CPU, and duration information. Note that this view is updated only when a query is completed, so you might need to retrieve information multiple times if your server currently has a large workload. As in Listing 2, you can replace the asterisk by the list of columns that you're particularly interested in.

**Metadata Everywhere**

We've delved further into the metadata that keeps track of and displays information about your queries and their execution plans. I showed you the views and functions available and how they could be combined with each other to give different sets of information. If you explore the metadata in more depth, experimenting with all the information provided in the columns contained in the output, you'll be able to determine which views work best in your environment and can help you find the performance data most useful to your troubleshooting endeavors.

SQL

InstantDoc ID 97975

**LEARNING PATH****Learn more about plan caching:**

- "Examining XML Plans in Cache,"  
InstantDoc ID 97562
- "Inside SQL Server: SQL Server 7.0 Plan Caching,"  
InstantDoc ID 5915
- "Managing Plan-Cache Size,"  
InstantDoc ID 97373
- "Troubleshooting Parallel Queries on  
SQL Server 2005,"  
InstantDoc ID 92661

**Learn more about XML query plans:**

- "SQL Server 2005 Management Tools,"  
InstantDoc ID 46798
- "The XML Query-Plan Treasure Trove,"  
InstantDoc ID 95498

**Learn more about SQL Server internals:**

- "Why Learn About Internals?"  
InstantDoc ID 93143

**Learn more about these dynamic management views:**

- sys.dm\_exec\_text\_query\_plan, [technet.microsoft.com/en-us/library/bb326654.aspx](http://technet.microsoft.com/en-us/library/bb326654.aspx)
- sys.dm\_exec\_requests, [technet.microsoft.com/en-us/library/ms177648.aspx](http://technet.microsoft.com/en-us/library/ms177648.aspx)
- sys.dm\_exec\_query\_stats, [msdn2.microsoft.com/en-us/library/ms189741.aspx](http://msdn2.microsoft.com/en-us/library/ms189741.aspx)

**For detailed troubleshooting, you can use sys.dm\_exec\_query\_stats to return performance information for individual queries.**

# Sharpen Your Basic SQL Server Skills

## Surface Area Configuration tools reduce exposure to security risks

**Q: What's the SQL Server surface area, and what security risks are associated with it?**

**A:** All SQL Server components that are accessible by users or applications are known as the surface area. Enabling all of SQL Server's features increases the risk of a security breach by unauthorized users or applications. To make the vulnerabilities of the surface area less visible to potential attackers, the default installation of SQL Server 2005 disables most of the at-risk features (e.g., xp\_cmdshell, dedicated administrator connection). You can also deactivate unused components to reduce surface area exposure.

**Q: What tools are available to increase the security of the SQL Server surface area?**

**A:** You can use two SQL Server 2005 Surface Area Configuration tools to increase surface area security and manage common service-related operations. These tools use Window Management Instrumentation (WMI) API calls to manage SQL Server's registry operations. The Surface Area Configuration for Services and Connections tool gives you a simple method for configuring the security of local, as well as remote, SQL Server machines. For more information, see [msdn2.microsoft.com/en-us/library/ms188980.aspx](http://msdn2.microsoft.com/en-us/library/ms188980.aspx). With the Surface Area Configuration for Features tool you can enable and disable features of the Database Engine, SQL Server Analysis Services (SSAS), and SQL Server Reporting Services (SSRS). For more information, see [msdn2.microsoft.com/en-us/library/ms183753.aspx](http://msdn2.microsoft.com/en-us/library/ms183753.aspx).

**Q: What features does the Surface Area Configuration for Features tool include?**

**A:** The features in the Surface Area Configuration for Features tool are divided into three main categories: SSAS, Database Engine, and SSRS. SSAS features include Ad Hoc Data Mining queries, Anonymous Connections, Linked Objects, and User-Defined functions. Database Engine features include Ad Hoc Remote Queries, Common Language Runtime (CLR) Integration, Database Mail, OLE Automation, Service Broker, SQL Mail, Web Assistant stored procedures, xp\_cmdshell, Dedicated Administrator Connection, and Native XML Web Services. SSRS features include HTTP access and Web Service Requests, Scheduled Events and Report

Delivery, and Windows Integrated Security.

**Q: What database administrative services does the Surface Area Configuration for Features tool support?**

**A:** The tool provides an interface from which you can start, stop, pause, or resume administrative services. You can also configure whether the startup is manual, automatic, or disabled. In addition to SSAS, SSRS, and SQL Server Integration Services (SSIS), these configuration services are supported:

- Database Engine
- Full-Text Search service
- SQL Server Integration Services (SSIS)
- MSSQLServerADHelper service
- Notification Services
- SQL Server Agent service
- SQL Server Browser service
- SQL Server Writer Service

**Q: How do I use the Surface Area Configuration for Features tool?**

**A:** You can use the tool's GUI, T-SQL in the Query Editor, or a command-line utility. To use the GUI go to Start, All Programs, SQL Server 2005, Configuration Tools, SQL Server Surface Area Configuration for Features. You can use the GUI to select a component and configure its features. To use T-SQL in the SQL Server Query Editor you need to enable xp\_cmdshell, as the following example shows.

```
sp_configure 'show advanced
options', 1
RECONFIGURE
GO
sp_configure 'xp_cmdshell', 1
RECONFIGURE
GO
```

Use the command-line sac utility when you're configuring multiple SQL Server machines with the same import and export Surface Area Settings. This utility creates a configuration file on the local server; this file is then exported to another server where the system imports it. The sac utility is in the \Program Files\Microsoft SQL Server\90\Shared folder. **SQL**

InstantDoc ID 98299



**Pinalkumar Dave**

([pinal@sqlauthority.com](mailto:pinal@sqlauthority.com)) is a database administrator and project manager. He writes regularly about SQL Server technology at [www.sqlauthority.com](http://www.sqlauthority.com).



# Superior Data Visualization Technology For Today (& Tomorrow)



# Dundas

Data Visualization

## Get the most out SQL Reporting Services!

If you need to add new data visualization and digital dashboard capabilities to your reports within **SQL Server 2005 Reporting Services**, then look no further than Dundas.

**Dundas Chart, Dundas Gauge, Dundas Map** and **Dundas Calendar** for Reporting Services offer unique, seamlessly integrated data visualization functionality for SQL Server 2005 Reporting Services. These four products make up the Dundas Dashboard Bundle for Reporting Services, giving you all you need to turn your reports into visually appealing dashboards and scorecards - quickly and easily.

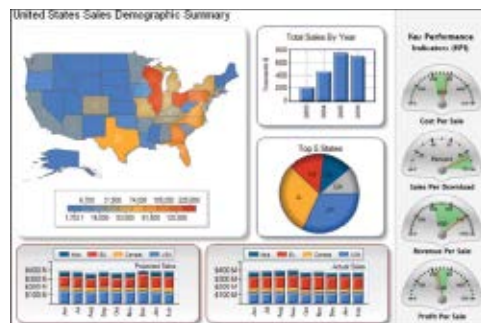
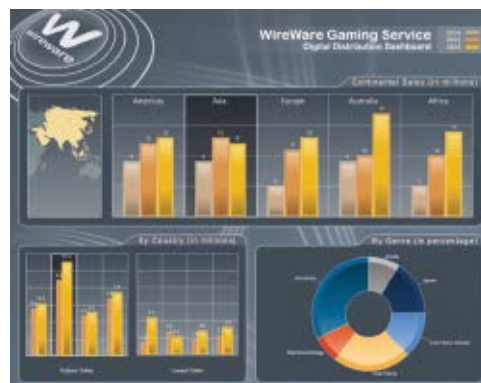
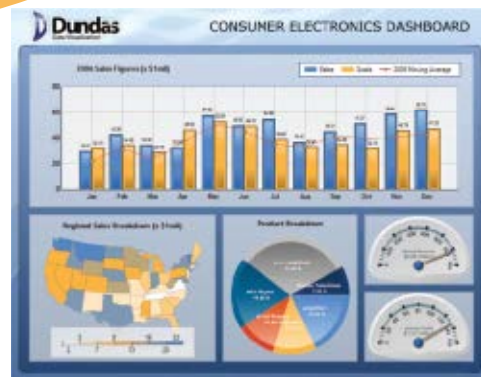
And the projects you develop today with Dundas technology for SQL Reporting Services 2005 can be migrated to SQL Server 2008.

Download **full evaluation copies** today and add advanced business intelligence to your corporate reports.



[www.dundas.com](http://www.dundas.com)

Microsoft and SQL Server are registered trademarks of Microsoft Corporation in the United States and/or other countries.



**Microsoft**  
**GOLD CERTIFIED**  
Partner

Data Management Solutions

[www.dundas.com](http://www.dundas.com)  
[info@dundas.com](mailto:info@dundas.com)  
(416) 467-5100  
(800) 463-1492

**Advanced Data Visualization  
for Microsoft® Technologies**

## Data Warehousing:

# Horizontally Partitioning the Fact Table

Manage large tables by dividing them into discreet, non-overlapping sections

**F**act tables tend to grow very large, very fast. Sometimes, a fact table can become so large that it can be difficult to manage, and SQL queries can be adversely affected by the sheer size of the data set involved. However, you can horizontally partition a fact table to make it easier to work with.

A horizontally partitioned table is one in which the rows are divided into discreet, non-overlapping sections. Each section in a horizontally partitioned table is defined by a range of values, such as by date, geographic area, or customers' last names. (Note: You might consider storing each section on a different hard disk to enhance query performance.) Although it isn't difficult to horizontally partition a table, it does require some advance planning because you're dealing with large amounts of data. Let's explore when and why you'd want to horizontally partition a fact table, and how to create a partition function, a partition scheme, and a partitioned table using SQL Server 2005's built-in functionality.

### Why Partition a Fact Table?

Large tables (i.e., tables with hundreds of millions of rows) can be difficult to manage because of their size and the amount of time it takes to do anything with them (e.g., rebuild an index). In a transactional database, the associative tables (i.e., those tables that involve the many to many—M:N—relationship) are often the tables with the most rows in the database. In dimensional modeling, a fact table is the equivalent of an associative table. Like an associative table in a transactional database, a fact table often has many more rows than its related dimensions, perhaps even as many as (# of rows in dimension 1) x (# of rows in dimension 2) ... x (# of rows in dimension *n*) rows.

Partitioning breaks these monster tables into manageable chunks. If your maintenance time windows are shrinking or the amount of data to be processed is growing, you can partition the table and perform

tasks, such as backup and restore operations or use the Database Contingency Checker (DBCC) to update table statistics, by partition instead of for the entire table. SQL Server 2005 treats the many sections of a partitioned table as a single logical entity, and the multiple partitions appear as a single table to end users.

The following are some reasons to horizontally partition a table:

- You can better control where each partition is placed in storage and leverage multiple read/write heads for fast query resolution.
- You can back up and restore by partition, indexes can be rebuilt and reorganized by partition, and the indexes themselves can be partitioned.
- You can direct queries that include a WHERE clause that contains either the partitioning column or an indexed column to the appropriate partition for resolution.
- You can reduce lock escalations and lock-management overhead because locking is limited to partitions.
- You can merge or split partitions fairly easily if multiple partitions are in the same file group.

So which tables are the best candidates for horizontal partitioning? Very large tables, tables that you expect to grow very large in the near future, and tables that can be intuitively partitioned based on their business value (e.g., by fiscal year). These tables must include a column whose values are NOT NULL and that can be used to divide the rows into discreet, non-overlapping sections such as a column containing sales dates.

If your database contains a large table against which queries and updates aren't performing the way you think they should, consider testing how partitioning might affect query performance in your environment. SQL Server 2005 is partition-aware, meaning that if slow-running queries include a WHERE clause that contains the partitioning column or the indexed



**Michelle A. Poollet**

(mapoollet@sqlmag.com) is a contributing editor for *SQL Server Magazine* and cofounder of Mount Vernon Data Systems. She teaches database design and programming, has a master's degree in computer information systems, and is an MCP.



### Author's Tip

If you recently migrated to SQL Server 2005 from SQL Server 2000 and had been using partitioned views, you should consider replacing the partitioned views with partitioned tables. However, you'll need to test these changes in a controlled environment to determine if you should, in fact, replace the partitioned views with partitioned tables.

—Michelle A. Poollet

```

SALES Fact
SalesFact_KEY (PKEY)
Date_of_Event
Time_of_Event
SalesTransactionKEY (FKEY)
SalesAssociateKEY (FKEY)
CustKEY (FKEY)
ProductKEY (FKEY)
StoreKEY (FKEY)
GeographicAreaKEY (FKEY)
...and other columns...

```

Figure 1

The SALES fact table

column (and the index is also partitioned), only the relevant partition is accessed for query resolution. This functionality can significantly help performance.

### Creating a Partition Function

To partition a table, you need to use a function that's composed of a partitioning column and a set of boundaries. To create the partition function for the SALES fact table shown in Figure 1, you'd run the command

```

CREATE PARTITION FUNCTION
MyPartitionFunctionLeft
(datetime)
AS RANGE LEFT
FOR VALUES ('1/01/2003',
'1/01/2005', '1/01/2007')

```

MyPartitionFunctionLeft is the name of the partitioning function, (datetime) is the data type of the partitioning column, and RANGE LEFT stipulates how to divide up the data values that are bound by the FOR VALUES dates.

There's two ways to set the boundaries: RANGE LEFT or RANGE RIGHT. The RANGE LEFT clause divides the data from the lowest value to the highest value (i.e., in ascending order). The RANGE RIGHT clause divides the data from the highest value to the lowest value (i.e., in descending order).

The partitioning column is often a datetime data type such as the Date\_of\_Event column (shown in Figure 1). Separating datetime records into non-overlapping groups is straightforward. For example, if your business rules and known operational queries indicate that partitioning the table on the date of a sales event is reasonable, then you could partition the data into two-year groupings, as I did in the previous partition function command. Partitioning RANGE LEFT divides the data into the value ranges shown in Figure 2.

P1 <= 1/01/03	P2 > 1/01/03, <= 1/01/05	P3 > 1/01/05, <= 1/01/07	P4 > 1/01/07
------------------	--------------------------------	--------------------------------	-----------------

Figure 2

Partitions created with the RANGE LEFT clause

If the date of a sales event was June 23, 2004, you'd find that record in partition 2 (P2). If you want to create the partition function with the RANGE RIGHT clause, you'd run the command

```

CREATE PARTITION FUNCTION
MyPartitionFunction datetime
AS RANGE RIGHT
FOR VALUES ('1/01/2003',
'1/01/2005', '1/01/2007')

```

Partitioning RANGE RIGHT divides the data into the value ranges shown in Figure 3. I recommend, for the sake of consistency and for the ease of querying, that you choose one range declaration (i.e., RANGE LEFT or RANGE RIGHT) and stick with it throughout all the partitioned tables that you create in your environment.

Each range of values in a partition is restricted by boundaries that are specified in the FOR VALUES clause. Note that if you're using datetime data types for boundary values and your company has an office in Europe, you'll have to decide on an international standard for datetime, so that it's uniform across your company. SQL Server assumes that *us\_english* is the default language for the session, so if that's not the case, you'll want to create a user-defined function (UDF) that will convert various date formats into *us\_english*, and reference that UDF in the FOR VALUES clause. You don't have to use literals in the FOR VALUES clause; you can reference variables, functions, and UDFs.

### Creating a Partition Scheme

Now that you've created a partition function, you need to create a partition scheme. The partition scheme maps partitions to various file groups, as shown in the following command:

```

CREATE PARTITION SCHEME
MyPartitionScheme
AS MyPartitionFunction
TO (MyFilegroup1, MyFilegroup2,
MyFilegroup3, MyFilegroup4,
MyFilegroup5)

```

MyPartitionScheme is the name of the partitioning scheme, and MyPartitionFunction refers to the partition function. This command maps the boundary values into partitions that are then assigned to one or more file groups. Data rows with Date\_of\_Event datetime values prior to 1/01/03 are assigned to MyFilegroup1. Rows with datetime values greater than or equal to 1/01/03 and prior to 1/01/05 are assigned to MyFilegroup2. Rows with datetime values greater than or equal to 1/01/05 and prior to 1/01/07 are assigned to MyFilegroup3. All other rows with datetime values greater than or equal to 1/01/07 are assigned to MyFilegroup4.

For each set of boundary values (remember, boundary values are in the FOR VALUES clause of the partition function statement), there will be (# of boundary values + 1) partitions. The previous CREATE PARTITION SCHEME statement includes three boundary values and four partitions. No matter whether the partitions are created RANGE RIGHT or RANGE LEFT, there will always be (# of boundary values + 1) partitions, up to 1000 partitions per table. So why are there five file groups instead of four in this



#### Editor's Tip

Remember, every partition function and partition scheme name has to be unique within a database.

—Megan Bearly, associate editor





example? (Remember, the partition function statement had only three boundary values.) The fifth file group is the optional “next-used” file group. Let’s look at how this file group is used and how important it is in the partition scheme.

In the `CREATE PARTITION FUNCTION` example, the last boundary value is 1/01/2007, so records with a `Date_of_Event` value greater than or equal to the 1/01/2007 will be stored in partition 4 (P4). When 1/01/2009 rolls around, you’ll have to create a new partition to maintain the storage scheme that you’ve established. If your original `CREATE PARTITION SCHEME` statement didn’t include a next-used file group, you wouldn’t be able to split P4 into P4, which will hold data from 2007 through the end of 2008, and partition 5 (P5), which will hold data from 1/01/2009 and beyond. If your partitioning plan involves periodically creating new partitions to hold new data, as in this example, make sure you include the next-used file group in your `CREATE PARTITION SCHEME` statement.

You don’t have to assign one partition to one file group, as I did in Figure 4. Instead, you can map multiple partitions to a single file group or you can map all the partitions to one file group. However, you can’t map one partition to multiple file groups.

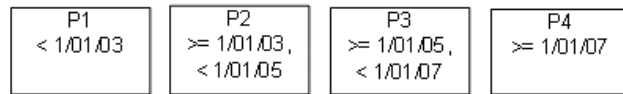
Creating the partition scheme might be the most important step in the partitioning process. In the future, you might want to combine the data from two adjacent partitions into one partition, add a boundary value to an existing partition, or move data from a populated partition into an empty partition. To perform these operations, you’ll want to do some advance planning and create the partition scheme so that it will support these activities. SQL Server 2005 Books Online (BOL) provides some good planning resources, including “Planning Guidelines for Partitioned Tables and Indexes” ([msdn2.microsoft.com/en-us/library/ms180767.aspx](http://msdn2.microsoft.com/en-us/library/ms180767.aspx)).

## Creating a Partitioned Table

Creating a partitioned table isn’t all that different from creating a regular table; you just have to reference the name of the partition scheme in the `ON` clause. To do so, run the command

```
CREATE TABLE SALESFact
(SalesFact_KEY      bigint identity
(1, 1) primary not clustered NOT
NULL,
...all the other columns in the
table, ending up with the
partitioning column...,
Date_of_Event      datetime
NOT NULL)
ON MyPartitionScheme (Date_of_Event)
```

By specifying the name of a partition scheme, you’re indicating that this table is a partitioned table. Obviously, the partition scheme and the partition function have to exist in the database before you can create the table.



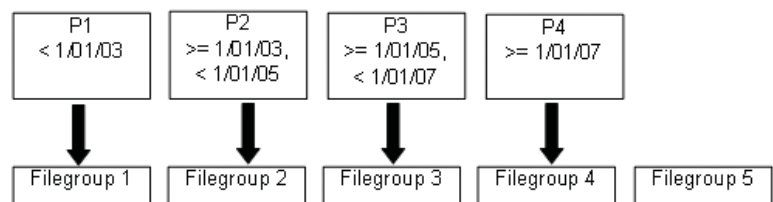
**Figure 3**

Partitions created with the `RANGE RIGHT` clause

You can combine the data from multiple partitions into one partition. However, you can only merge two adjacent partitions at a time, so if you’re trying to “unpartition” a table that has multiple partitions, you’ll have to repeat this step many times. To merge two partitions, run the command

```
ALTER PARTITION FUNCTION
MyPartitionFunction()
MERGE RANGE ('1/01/2003')
```

This command will merge the partitions as follows: partition 1 (P1) will merge into P2, meaning that P2 will then contain all rows with a `Date_of_Event` value prior to 1/01/05. Internally (i.e., in the `sys.partitions` system table), the partitions will be renumbered, starting with one (not zero). P1 and P2 will become P1, P3 will become P2, and P4 will become P3. I recommend sketching out the DDL before you merge any partitions because you could potentially cripple your operations for sustained periods of time if you’re not careful when you perform these merges. In fact, Eric Hanson, Microsoft’s lead program manager for query processing on the SQL Server Relational Engine Team, recommends performing splits and merges only on empty partitions, even if that means you have to temporarily empty a partition.



**Figure 4**

Mapping partitions to file groups

## Easily Manage Monster Tables

SQL Server 2005’s ability to horizontally partition a table into non-overlapping sections and place each section on a separate partition makes it easy to manage large fact tables. Because the SQL Server 2005 Enterprise and Developer Editions are partition-aware, properly-written queries will access only the rows within the relevant partition, thus running much faster than if they had to traverse the contents of the entire table.



InstantDoc ID 98007

# 3 Info-packed eLearning seminars for \$99!

## EXCHANGE 2007 Mastery Series—Part 2

Hosted by WindowsIT Pro

### WHEN

May 29, 2008—11:00 AM, EDT

### WHERE

On your computer

### COST

\$99/registrant for 1, 2, or all 3 live online sessions, and includes access to the archived versions

### SESSIONS

- **Mailbox High availability in Exchange 2007: Learn the Pros and Cons of Your High-Availability Options**
- **Transport Rules: See Real-World Examples You Can Implement in Your Environment**
- **PowerShell: Get Started with Basic Commands—You Don't Need a PhD in Rocket Science**

**RESERVE A SEAT by going to:**  
[www.windowsitpro.com/go/elearning/MasteringExchange2007](http://www.windowsitpro.com/go/elearning/MasteringExchange2007)

### SPEAKER

Mark Arnold

MCSE+M, Microsoft MVP



Mark Arnold is a senior technical architect for Anix, a UK-based storage integrator, where he solves storage and compliance problems for his clients by using Microsoft Exchange as a key component in SAN and NAS deployments. He's also a regular contributor to Microsoft's "Industry Insiders" TechNet program and is active on Exchange newsgroups and forums.

### ABOUT THE SESSIONS

#### **Mailbox High Availability in Exchange 2007: Learn the Pros and Cons of Your High-Availability Options**

Exchange 2007 now has several acronyms for high availability—LCR, CCR, SCR—not including anything you can do with your storage or CDP solutions. Which method is best for you? How can you implement a mix of options to make your environment highly available at a price point that doesn't break the bank?

#### **Transport Rules—Real-World Examples That Can Help Your Environment**

What are transport rules and how do they help you administer your Exchange environment? You can find many complicated—and largely useless—examples on the Internet. We'll show some interesting things you can do with message flow and give you real-world examples.

#### **PowerShell—The Things You Need That Don't Involve a PhD in Brain Surgery or Rocket Science**

Some Exchange admins resist PowerShell because they think they can complete tasks quicker through the GUI. But we'll present some useful, quick, and readily repeatable PowerShell commands that will make your job easier rather than your hair grayer.

**REGISTER TODAY — seats are limited,**  
to allow lots of live Q&A at the end.

For more information, or to register, go to:  
[www.windowsitpro.com/go/elearning/masteringexchange2007](http://www.windowsitpro.com/go/elearning/masteringexchange2007)

# WindowsIT Pro

# SQL Server Backup Compression Shootout

3 tools offer similar compression performance but varied feature sets

**Editor's Note:** This summary of the comparative review presents the overall findings. For the complete review of each product, go to InstantDoc ID 98180 at [www.sqlmag.com](http://www.sqlmag.com).

The market for SQL Server backup compression tools has become crowded. Prior to the availability of compression tools such as Quest Software's LiteSpeed, many DBAs were using methods such as ZIP compression on their SQL Server backup files. But previous methods are inefficient and don't eliminate the need for an original, large backup; they also consume nearly double the disk space (and time) while the ZIP file is being generated. Today, databases are much larger than ever before and more mission-critical. How do you back up a 500GB, 1TB, or larger database effectively and quickly? Then, how do you copy that large backup to another location rapidly? The short answer is that you can't. Both backup and copy operations can take a considerable amount of time.

The benefits of compression are obvious: A backup file takes up less physical disk space (reducing storage

costs), and when archived needs less space on tape or in your vault. Smaller files mean faster copies and quicker restore times. Another reason for the increased importance of backup compression tools is compliance with regulations such as the Sarbanes-Oxley (SOX) Act and the Health Insurance Portability and Accountability Act (HIPAA). Although the cost of disks has come down over the years, disks can't solve all the storage problems related to larger data sets.

## Backup Compression and You

A backup compression tool's main purpose is to compress a backup as it's being made and give you the ability to restore it. Individual tools add their own feature sets on top of that basic functionality. All integrate with SQL Server in different ways, and all are relatively inexpensive. Choosing the program that's right for you comes down to evaluating features, effectiveness, cost, and ease of use. Ask the following questions as you evaluate compression tools:

- Are there backup problems that a compression tool will solve, such as the inability to retain backups on disk or a lack of disk space for backups?
- Are there immediate cost savings that you can realize with compressed backups (e.g., when the internal storage team is charging for storage space by the gigabyte)?



**Allan Hirt**

([allan@sqlha.com](mailto:allan@sqlha.com)) consults for and trains clients all over the world. He has written or co-authored numerous SQL Server publications; his latest book is *Pro SQL Server 2005 High Availability* (Apress). You can find him on the Web at [www.sqlha.com](http://www.sqlha.com).

## MORE on the WEB

Read the Web-exclusive sidebar at InstantDoc ID 98178.



## HYPERBAC FOR SQL SERVER

**Pros:** Excellent integration with SQL Server; no updating of SQL Server Agent jobs or T-SQL scripts; simple installation; ZIP functionality; compression of bcp, DTS, and SQL Server Integration Services (SSIS) files to disk; fastest restore of products compared

**Cons:** No GUI-based backup tools

**Rating:** ★★★★★☆

**Price:** \$699 per server

**Recommendation:** HyperBac for SQL Server matches the way many DBAs like to work and integrates with SQL Server seamlessly out of the box. Coupled with innovative features such as backing up to a .zip file and the ability to compress bcp, DTS, and SSIS streams to and from files, HyperBac for SQL Server is a clear winner.

**Contact:** HyperBac Technologies • [www.hyperbac.com](http://www.hyperbac.com)



## SQL BACKUP

**Pros:** Above-average management tool; solid performance; Compression Analyzer feature

**Cons:** 64-bit, encryption, and cluster support only in Pro version; problems with restore operation; inaccurate compression savings

**Rating:** ★★★★★☆


**Price:** \$795 per server for Pro version; \$295 per server for Lite version; \$395 for SQL Data Compare tool

**Recommendation:** SQL Backup is a solid all-around offering with an excellent management tool, albeit one that has shortcomings such as the inability to kill connections. The installation process could be streamlined, and 64-bit support in the Lite version is something Red Gate should consider.

**Contact:** Red Gate Software • [www.red-gate.com](http://www.red-gate.com)



TABLE 1: Features Comparison

	 HyperBac for SQL Server	SQL Backup	SQLsafe
SQL Server Versions Supported	7.0, 2000, 2005	2000, 2005	7.0, 2000, 2005
x86 (32-bit) Support	Yes	Yes	Yes
x64 (64-bit) Support	Yes	Yes (Pro version only)	Yes
IA64 (64-bit) Support	Yes	Yes (Pro version only)	Yes
Supports Clusters	Yes	Yes (Pro version only)	Yes
Requires Changing of Existing Backup Processes, Scripts, and Jobs	No	Yes	Yes
GUI-based Tool	No	Yes	Yes
Command-Line Tool	No	Yes	Yes
Extended Stored Procedures for Scripting	No	Yes	Yes
Other SQL Server Compression	Yes: bcp, DTS/SSIS (writing to files only), xcopy of mdf/ldf	No	No
Backs Up Direct to ZIP	Yes	No	No
Conversion Tool for Backups to Native SQL Backup	Yes	Yes	Yes
Variable Compression Rate	Yes (2 levels)	Yes (3 settings)	Yes (4 levels plus 2 Intelli-Compress settings)
Granular/Object Level Restore	Object level recovery with additional feature (Backup Explorer)	Row-level restore requires additional product: SQL Data Compare	Table-level restore
Encryption	AES up to 256-bit	128- or 256-bit AES (Rijndael); 256-bit Pro version only	DES, Triple DES, RC2, Rijndael
Direct Integration with Other Backup Tools	No	No	Yes: IBM Tivoli

- Are there problems with your current disaster recovery plan in regard to backups that smaller, more agile backup files could help solve?
- Does the tool under consideration offer anything beyond compression that you can use in your environment, or is compression good enough?
- Does the licensing model cover the number of SQL Server servers and instances in your environment?
- How will changing the way backups are done affect your environment? How long will changes take to implement?

In this article, I compare three popular SQL Server backup compression tools: HyperBac Technologies' **HyperBac for SQL Server**, Red Gate Software's **SQL**



## SQLSAFE

**Pros:** Table-level restore; ability to create backup policies; freeware edition (with limited functionality) available

**Cons:** Highest cost; backups made outside of SQLsafe aren't tracked; slowest restore in testing

**Rating:** ★★☆☆☆

**Price:** \$995 per instance; SQLsafe Freeware edition available

**Recommendation:** I was disappointed with SQLsafe, especially since its features make it appear enterprise-ready. Its restore time was the slowest of the three tools tested, and the UI is great only if you love Outlook. It doesn't poll an instance to determine whether a backup was made outside of SQLsafe, which is a serious flaw. Some of its features don't seem to be supported in SQL Server 2005.

**Contact:** Idera • [www.idera.com](http://www.idera.com)

**Backup**, and Idera's **SQLsafe**. (Table 1 lists the products' feature sets.) You might be wondering why LiteSpeed for SQL Server isn't included here. Because the newest version of LiteSpeed for SQL Server (version 5.0) wasn't complete in time to be tested along with the other products, Quest opted not to participate in this comparison. (Quest is adding new features to LiteSpeed for SQL Server, including virtualized backups, support for non-SQL Server platforms, and policy-based management. *SQL Server Magazine* will review the updated LiteSpeed for SQL Server in a subsequent issue.) To read my take on how SQL Server 2008 will change the backup compression landscape, see the Web-exclusive sidebar "Backup Compression and SQL Server 2008," InstantDoc ID 98178. For a complete review of each product, go to InstantDoc ID 98180.

### The Tool That's Right for You

When it comes to disaster recovery, refreshing a development server with new data, or any operation that requires a restore, time is money. All three tools both sped up and compressed backups and had faster restore times than native SQL Server operations. Time spent backing up and compressing the backup was similar among all three products. HyperBac for SQL Server and SQL Backup had similar times for a restore operation; although SQLsafe's restore time was faster than a native SQL Server restore, it was considerably slower than the other two products' restore times. (To read more about my compression testing, see the sidebar "Compression Testing and Results.")

HyperBac for SQL Server is available with a per-server license model and also has the lowest list

## Compression Testing and Results

To test compression in the three products in this review, I used a 539GB real-world database loaned specifically for my testing. (A small database would not have yielded results showing how efficient a compression was.) The native backup file size that the 539GB backup generated was 440.53GB.

The server I used ran Windows Server 2003 Enterprise Edition x64 with 4GB of memory and had an AMD Opteron 144 1.8GHz single core processor. The disk configuration was two drives: one 250GB drive for the OS, and one 1TB Seagate Barracuda ES.2 SATA drive for the backups and the data to simulate many clients where the backups and data live on the same set of disks. I used SQL Server 2005 Enterprise Edition x64 with SQL Server 2005 Service Pack 2.

I generated all results by using default settings for the fastest time according to how each tool defines that time. All compression tools will add some amount of CPU overhead. This number will vary depending on your systems, but it's an important measurement to consider. You should run any tool you evaluate on your systems against your databases to get an accurate idea of how it will work on your standard server configurations.

Table A shows the test results. The asterisk next to the native SQL Server times denotes that they are estimated. I restored the database for the test from an external USB drive to the 1TB drive. Making a full backup on the 1TB drive wasn't possible because the actual database files took up about 540GB (formatted, the 1TB drive has 931GB of usable space).

To estimate the backup time, I backed up one of the 200GB files that made up the database, then restored it. The backup

took 3:20:21, was 179.75GB, and had a throughput of 16.051MB/sec. The restore took 3:07:19 with a throughput of 17.167 MB/sec. To get the numbers listed in the table for the native timings, I extrapolated the data, knowing how large the entire backup file was. What I did underscores the reason behind most of these compression tools: There comes a point when the database gets so large that you don't have enough space to make a full backup on your drive, so you get creative (e.g., by making file backups, shrinking data or log files, and so on).

InstantDoc ID 98179

**TABLE A: Test Results**

	Native SQL Server Backup	HyperBac for SQL Server	SQL Backup Pro	SQLsafe
Version Tested	9.00.3042	2.7.0.2	5.2.0.2825	4.6.312.4023
Backup Time (hours:minutes:seconds)	7:48:00*	3:33:21	2:59:37	2:53:24
Backup Size (GB)	440.53	167.54	167.57	166.12
Compressed File Size vs. Original Backup	N/A	61.97%	61.96%	62.29%
Restore Time (hours:minutes:seconds)	7:20:00*	4:01:26	4:06:05	5:58:34

price without reduced functionality. Imagine that you are administering 100 instances. If you spread those instances over 65 physical servers, the list-price comparison among the products would be \$99,500 for SQLsafe, \$51,675 for SQL Backup Pro, and \$43,875 for HyperBac for SQL Server. Despite the cost differential, Idera deserves credit for not requiring a license for the SQLsafe agent on instances that are performing only restores, which is perfect for disaster recovery.

Where workflow is concerned, HyperBac for SQL Server is at the top of the list based on the way most DBAs I know like to work. Although it doesn't ship with a management tool for backups, it works with whatever tool you have. As I've noted, a big part of the pain of tool adoption is the cost of implementation. HyperBac wins hands down on this point because it lets you get up and running in short order. But if you're managing a lot of SQL Server backups and want the full complement of GUI, management tool, and the ability to use the command line or scripts via extended stored procedures, either SQL Backup or SQLsafe might be better bets for you. Both these products have a management tool that is geared toward environments with multiple SQL servers. Neither management tool is perfect, but either might do a better job for you than managing your backups in each instance with SQL Server Management Studio (SSMS).

Both SQLsafe and SQL Backup were more cumbersome to install than HyperBac for SQL Server, involving multiple steps such as pushing components out to SQL servers after the initial install. HyperBac for SQL Server's install could become part of a standard Windows build process so that it's already in place by the time a DBA gets a new server.

All three tools offer similar performance in compression, so an ultimate purchasing decision boils down to the other features and functionality a tool brings to the table and how it can benefit your own backup and restore administration. I found SQL Backup and SQLsafe both slightly disappointing; neither provides the ability to set restore policies or lets you manage regularly scheduled restores. In enterprise-class tools, these are glaring omissions. Using the cost example of 100 instances, if you compare according to price and feature set, is SQLsafe approximately two times better than either SQL Backup or HyperBac for SQL Server? In my opinion, the answer is no. Both HyperBac for SQL Server and SQL Backup Pro represent good value, and each takes a different approach. If I were in the market for a SQL Server backup compression tool, I'd choose HyperBac for SQL Server, and that's why I've designated it my Editor's Choice. But I recommend that you audition the tools that appeal to you and make your own informed choice. **SQL**

InstantDoc ID 98180



### Editor's Tip

Got a great new product? Send announcements to products@sqlmag.com

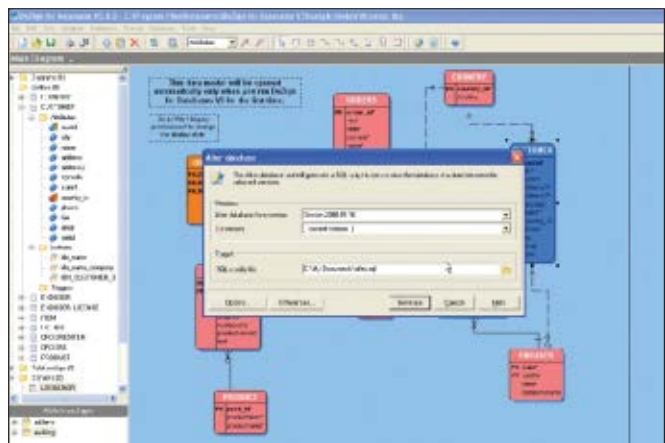


—Jeff James,  
senior editor

## DATABASE MODELING

### Create, Compare, and Update Visual Database Models

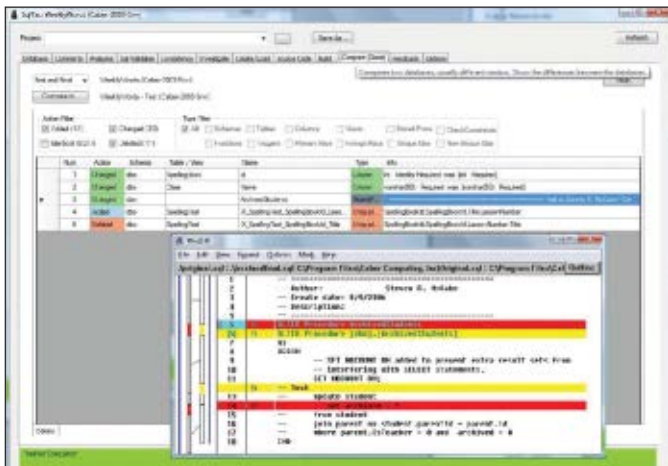
Datanamic Solutions B.V. has announced the release of **DeZign for Databases 5**, an application that allows DBAs to create, compare, and modify database models. According to Datanamic, the software relies on a visual creation metaphor that simplifies the creation of data models. This latest version introduces improved data and function conversion rules that improve accuracy when migrating data models from one platform to another. SQL Server 2005, 2000, 7.0, and 6.5 are supported, and pricing begins at \$245 per user. For more information, contact Datanamic Solutions at support@datanamic.com or visit [www.datanamic.com](http://www.datanamic.com).



## BUSINESS INTELLIGENCE

### OLAP Reporting and Analysis Tool

CNS International's **DataWarehouse Explorer** analyzes and generates reports from SQL Server, Oracle, and IBM databases. The software features an interface that closely resembles Microsoft Office 2007, a feature that helps end users more easily create their own reports, while an included server component enables DBAs to manage and view generated reports. DataWarehouse Explorer is written in .NET, and offers an OLAP reporting feature that requires the use of the Microsoft Analysis Services server. The application also relies upon the Microsoft Reporting Services server bundled with SQL Server 2005. For more information, contact Datanamic Solutions at sales@cns.nl or visit [www.cns.nl](http://www.cns.nl).



## DEVELOPMENT

### Database Programming and Maintenance

Caber Computing has released **SqlTac**, a product that provides a number of utilities for SQL Server DBAs and developers that rely on Visual Studio to create database applications. For DBAs, SqlTac can be used to validate SQL statements, capture domain knowledge, and troubleshoot and resolve generic database design issues. For developers, the product generates Dlinq's DataAccess layer, provides Object-Relational Mapping (ORM), and saves database schemas. It can also validate SQL code used in stored procedures and user-defined functions, and offers a variety of search tools for finding specific SQL statements. SqlTac supports SQL Server 2008, 2005, and 2000, and pricing begins at \$295 per user. For more information, contact Caber Computing Inc. at info@cabercomputing.com or visit [www.cabercomputing.com](http://www.cabercomputing.com).

## VIRTUALIZATION

### Flexible Product Release Management

Microsoft has announced the impending release of SQL Server 2008, the latest update to the SQL Server product platform. SQL Server 2008 will be the first product to use Microsoft's new virtualized product deployment platform, which separates product release dates from the real-world constraints of a rigid, predefined schedule. "With our new virtual product launch strategy, we can efficiently migrate our launch resources freely between different months, quarters, and even calendar years," explained a Microsoft spokesman. "Adhering to an arbitrary, inflexible date based on a legacy calendar first used more than 2,000 years ago is something we're moving away from. Our customers have asked for more flexibility as to when they receive new software, so this virtualized product deployment platform is the solution." The next product slated to use this new virtual product deployment system is the successor to Windows Vista, which has been allocated a 10-year virtual machine launch resource between the years of 2010 and 2020. **SQL**

InstantDoc ID 98199



Connecting readers with the products and services they need most.



## MarketPlace

A SPECIAL MONTHLY  
ADVERTISING SECTION

- Software • Training & Certifications • Career Opportunities • Consulting Services • Hardware & More!

If you would like your products and services featured among these pages, call:

### SQL Sales Manager

Birdie Ghiglione  
619-442-4064

birdie.ghiglione@penton.com

Additional advertising information is also available at [www.sqlmag.com](http://www.sqlmag.com)

**SQLUSA**  
**Best SQL Training**  
[www.sqlusa.com](http://www.sqlusa.com)

**Get data right - right from the start**

### AddressObject API

Verify, cleanse and format customer data at the point of entry or in batch. Easily integrate with .Net, MS SQL or Java.

#### Data Submitted

22342 empris ste 100 92688

#### Address Data Returned

22342 Avenida Empresa Ste 100  
Rancho Santa Margarita, CA  
92688-2156  
Carrier Route: C056  
CountyName: Orange  
TimeZone: Pacific  
Delivery Indicator: Business  
Suite Status: Validated  
Latitude: 33.6480  
Longitude: -117.6000  
(and more)

**Get a FREE demo at:**  
**MelissaData.com/sql**  
**1-800-MELISSA**

**MELISSA DATA**  
Your Partner in Data Quality

## Advertiser Index

<b>Acronis</b> ..... 10	<b>Idera</b> ..... Cover 4
<a href="http://www.acronis.com">www.acronis.com</a>	<a href="http://www.idera.com">www.idera.com</a>
<b>Altova</b> ..... 6	<b>InterCerve</b> ..... Cover 3
<a href="http://www.altova.com">www.altova.com</a>	<a href="http://www.sqlsentry.net">www.sqlsentry.net</a>
<b>ApexSQL</b> ..... 20, 26	<b>Melissa Data Corporation</b> ..... 47
<a href="http://www.apexsql.com">www.apexsql.com</a>	<a href="http://www.melissadata.com/sql">www.melissadata.com/sql</a>
<b>Business Objects</b> ..... 24	<b>Quest Software Inc.</b> ..... 4
<a href="http://www.businessobjects.com/CR2008/dev">www.businessobjects.com/CR2008/dev</a>	<a href="http://www.quest.com">www.quest.com</a>
<b>Dell</b> ..... Cover 2, 1	<b>Red Gate Software Ltd.</b> ..... 9
<a href="http://www.dell.com/SQLBI">www.dell.com/SQLBI</a>	<a href="http://www.red-gate.com">www.red-gate.com</a>
<b>Dell</b> ..... 16B	<b>SQLUSA</b> ..... 47
<a href="http://www.dell.com">www.dell.com</a>	<a href="http://www.sqlusa.com">www.sqlusa.com</a>
<b>Dundas Software Ltd.</b> ..... 38	<b>SQL Connections</b> ..... 16
<a href="http://www.dundas.com">www.dundas.com</a>	<a href="http://www.DevConnections.com">www.DevConnections.com</a>
<b>Embarcadero</b> ..... 2	<b>Windows IT Pro</b> ..... 42
<a href="http://www.embarcadero.com">www.embarcadero.com</a>	<a href="http://www.windowsitpro.com">www.windowsitpro.com</a>
<b>Fujitsu Computer Systems Corporation</b> ..... 30	
<a href="http://www.us.fujitsu.com/computers/PRIMERGY">www.us.fujitsu.com/computers/PRIMERGY</a>	



**Michael Otey**

(motey@sqlmag.com) is technical director for *Windows IT Pro* and *SQL Server Magazine* and coauthor of *SQL Server 2005 Developer's Guide* (Osborne/McGraw-Hill).

# SQL Server 2008 Webcasts

**W**ithout a doubt, SQL Server 2008 is the most feature-rich release of SQL Server yet, making the task of learning about it a challenge. Microsoft has come out with an impressive collection of Webcasts designed to help you learn about the vast array of functionality SQL Server 2008 provides. All told, there are 24 different SQL Server 2008 Webcasts. They're free for the watching at the Microsoft site: [www.microsoft.com/sql/2008/learning/webcasts.aspx](http://www.microsoft.com/sql/2008/learning/webcasts.aspx). Here are five I like:

## The Next Release of Microsoft SQL Server: Overview

If you're looking for a quick, high-level overview of the new features in SQL Server 2008, then this 65-minute Webcast is for you. In it, Dave Campbell, Microsoft Technical Fellow and a software architect for the SQL Server Storage Engine, discusses the goals behind the SQL Server 2008 release. It also offers demos of the major new features and an overview of the SQL Server 2008 development process.

## What's New in Microsoft SQL Server 2008—MS Learning Clinic

Technically, speaking this isn't a Webcast but a set of three E-learning courses for SQL Server 2008. The three courses cover the new features in SQL Server 2008 business intelligence (BI), database

development, and how to use SQL Server 2008 as an enterprise data platform.

## Rich Report Design with SQL Server 2008 Reporting Services

This one-hour Webcast covers the reporting creation capabilities provided with SQL Server 2008 and offers an in-depth look at the enhancements to Reporting Services Report Designer. It also demonstrates the new visualizations for charts and gauges as well as the updates to the end-user-oriented Report Builder tool.

## The Value of Business Intelligence with SQL Server 2008

Getting a handle on exactly how BI fits in with your relational database and existing business model is a challenge. This 90-minute Webcast shows how SQL Server 2008 can help you utilize BI within your company, and it highlights many of the new BI features in SQL Server 2008.

## Overview of SQL Server Availability Features and Upcoming Improvements

High availability is right at the top of every DBA's list of top priorities. In this one-hour Webcast, you'll learn about SQL Server 2005's high availability features such as clustering, database mirroring, and log shipping, and you'll see how Microsoft is enhancing each of these features in SQL Server 2008.

**SQL**

InstantDoc ID 98241

## YOUR SAVVY ASSISTANT The Missing Link to IT Resources

**I'm** a little disappointed in Michael. We offer so many great SQL Server 2008 resources on our Web site, and he points you to Microsoft Webcasts. Although I'd like to give him a good verbal bashing, if he'd remembered to reference our content, they wouldn't need me to. So instead, I'd like to thank him for that little piece of job security.

Check out these articles and videos to better understand SQL Server 2008 and the new features it includes:

"New Features in SQL Server 2008," InstantDoc ID 96526

"Auditing and Compliance Features in SQL Server 2008," InstantDoc ID 96529

"How Can I Secure Data? Security in SQL Server 2008," InstantDoc ID 98035

"Under the Hood: Performance Tuning SQL Server 2008," InstantDoc ID 98036

"Protecting Your Data: Mirroring in SQL Server 2008," InstantDoc ID 98037

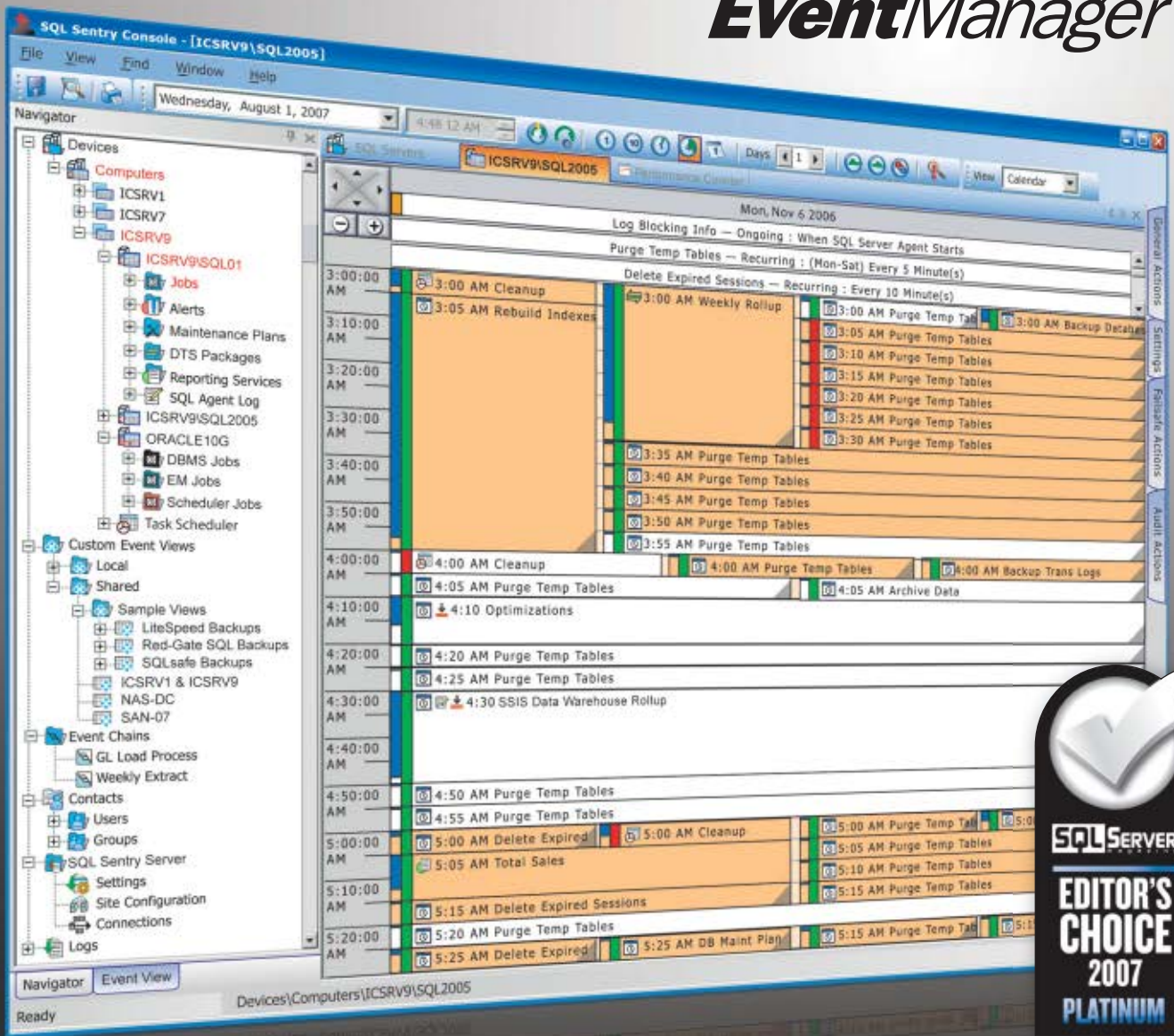
"Virtualization of SQL Server 2008," [www.sqlmag.com/go/VirtualizationofSQLServer2008](http://www.sqlmag.com/go/VirtualizationofSQLServer2008)



**Christian Humphries**

(christian.humphries@penton.com) is production editor for *Windows IT Pro* and *SQL Server Magazine*. She loves her dog. (A lot.)

*SQL Server Magazine*, April 2008, Vol. 10, No. 4 (ISSN 1522-2187). *SQL Server Magazine* is published monthly by Penton Media, Inc., copyright 2008, all rights reserved. SQL Server is a registered trademark of Microsoft Corporation, and *SQL Server Magazine* is used by Penton Media, Inc., under license from owner. *SQL Server Magazine* is an independent publication not affiliated with Microsoft Corporation. Microsoft Corporation is not responsible in any way for the editorial policy or other contents of the publication. *SQL Server Magazine*, 221 E. 29th St., Loveland, CO 80538, 800-621-1544 or 970-663-4700. Sales and marketing offices: 221 E. 29th St., Loveland, CO 80538. Advertising rates furnished upon request. Periodicals Class postage paid at Loveland, Colorado, and additional mailing offices. Postmaster: Send address changes to *SQL Server Magazine*, P.O. Box 447, Loveland, CO 80539-0447. Subscribers: Send all inquiries, payments, and address changes to *SQL Server Magazine*, Circulation Department, P.O. Box 447, Loveland, CO 80539-0447. Subscriptions in US, \$49.95 for one year; in Canada, US \$59, plus 6% for GST for one year; in UK £59 for one year; in all other countries, US \$99 for one year. Single-copy price in US, \$5.95; in Canada, US \$7.95. Printed in the U.S.A. BPA Worldwide Member.



## Unprecedented Visibility and Control

SQL Sentry Event Manager is the ultimate scheduling, alerting and response system for optimizing schedule performance of database servers across your enterprise. With Event Manager there are no agents to install and manage on each server. SQL Sentry Event Manager provides a visual display of SQL Agent jobs, Oracle DBMS Jobs, Oracle Enterprise Manager Jobs, Oracle 10g Scheduler, and Windows Tasks along with other events across the enterprise so that the DBA can "see" how they relate to one another and optimize the schedules more efficiently.

**Key Features:**

- Visual Schedule Management
- Alerting/Response System
- SSIS/DTS Support
- Cross-platform Support
- Chaining/Queuing
- Schedule Performance Monitoring

**Key Benefits:**

- Lower Database Administration Costs
- Improve Application Performance
- Reduce Downtime
- Distributed "Agent-less" Deployment
- 100% .NET Based

Free Trial Download at: [sqlsentry.net](http://sqlsentry.net)





# OVER 70,000 SQL SERVERS MONITORED WORLDWIDE



Over 70,000 can't be wrong. That's how many SQL Servers are monitored worldwide by Idera products.

**NEW SQL diagnostic manager** version 5.0 is a powerful solution that helps you monitor, diagnose, and analyze SQL Server performance across all SQL Servers in your environment—from a central console. And it proactively alerts you if a health, performance, or availability problem is detected.

- Monitor and manage SQL Servers enterprise-wide
- Find and fix performance bottlenecks
- Set customized alerts and notifications
- No agents or database objects required

## ON THE GO?

Monitor from your Blackberry or Windows Mobile Device too!

Download a **FREE** 14-day SQL diagnostic manager product trial:  
[www.idera.com/TheUltimate](http://www.idera.com/TheUltimate)

