



Global Knowledge®

Expert Reference Series of White Papers

Inside SQL Server
2008:
Management Studio
Improvements

Inside SQL Server 2008: Management Studio Improvements

Michael Manning, Global Knowledge Instructor

Introduction

When the SQL Server Management Studio (SSMS) was first introduced in SQL Server 2005, it received mixed reviews. Many DBAs have never totally embraced SQL Server Management Studio. With the launch of SQL Server 2008, Microsoft has succeeded in transforming the SSMS into a much more useful tool set. Microsoft has built in many new features to SSMS that make performing the routine DBA tasks easier and more intuitive to use. In fact, there are so many improvements to SSMS in SQL Server 2008 that it would take a small book to cover them all. SSMS at the core is still the same tool that we worked with in SQL Server 2005 but with many new features and improvements.

In addition to the new features, SSMS in SQL Server 2008 has benefited from performance gains as a result of faster communications between SSMS and the SQL Server engine. Besides the core SSMS features and improvements outlined in this paper, there are other hidden nuggets that you will find along the way as you start working with SQL Server 2008. This white paper will explore the core improvements to SSMS in SQL Server 2008.

SSMS 2008 Improvements

The following lists some of the key improvements in SQL Server Management Studio in SQL 2008.

- T-SQL Debugger
- Intellisense in the Query Editor
- Activity Monitor
- Object Explorer Details
- Object Search
- Multi-Server Queries

T-SQL Debugger

For programmers, it is very important to have a good debugging tool. Having said that, much of T-SQL code is written for maintenance tasks so debugging is not as critical as it is for application code. There are still many programmers, however, who write complex application code in T-SQL, so having a built-in query debugger is a huge benefit. The old Query Analyzer had a T-SQL debugger. SSMS in SQL Server 2005 had no such tool, the only option was to debug in Visual Studio. In SQL Server 2008, a debugging tool has been integrated into the Query Editor in the SQL Server Management Studio, making it easy to debug your T-SQL code. The debugger includes the ability to:

- Monitor values assigned to variables and parameters
- Step through T-SQL statements line by line
- Step through T-SQL statements using breakpoints
- Step into or over Stored Procedures, Functions, or Triggers

As with most Microsoft tools, there are usually several ways to achieve a task and the debugger is no different. To debug, simply add the code you want to a Query Editor window (see Figure 1), click the green debug arrow, click ALT-F5, or choose the "Start Debugging" option from the "Debug" drop down menu. The Query Editor will start the debugging process, and you can choose from the features listed above to step into and through your code to look for potential issues.

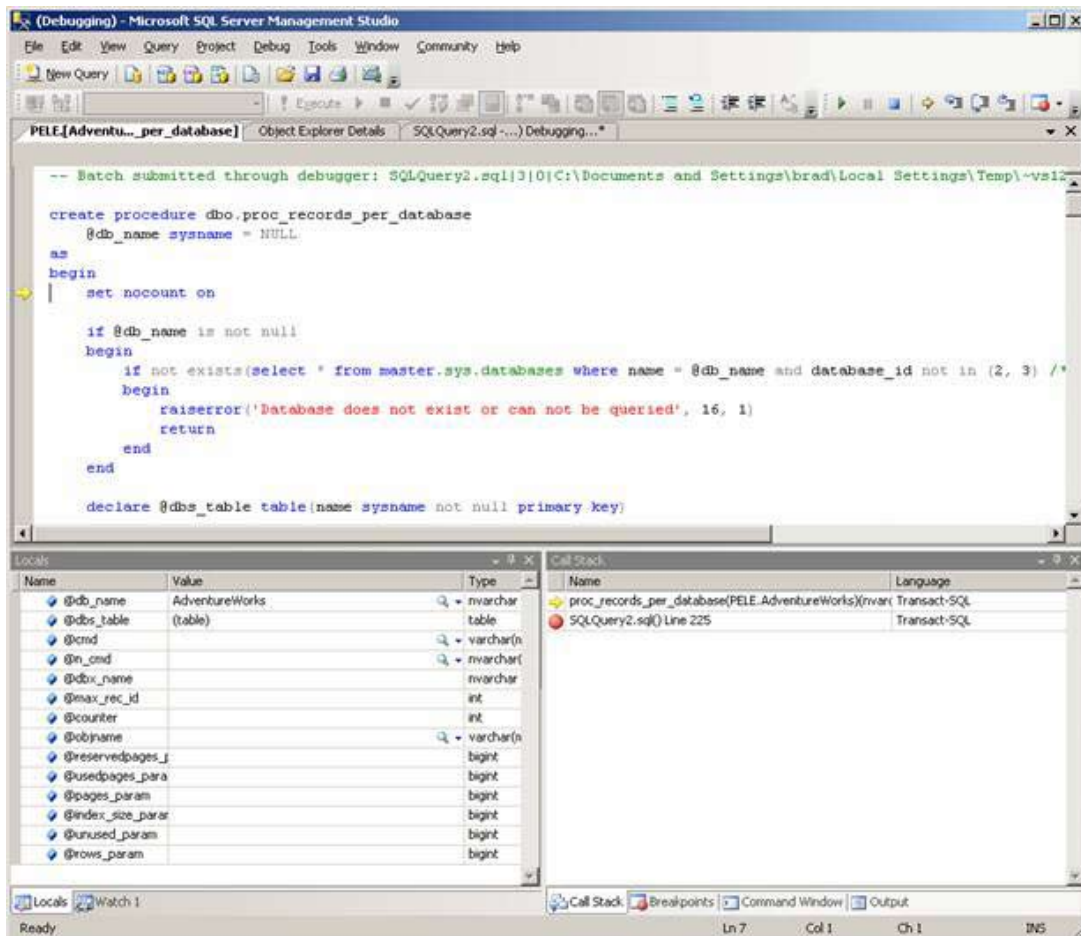


Figure 1: Debugging T-SQL

The T-SQL Debugger includes all core debugging features (the Locals, Watch, QuickWatch and Output windows). To step into stored procedures that are executed from the code, place a breakpoint at the line that calls the procedure and hit F11. You can also step into user-defined functions. Working with local instances of SQL Server

is relatively easy; working with remote instances can be a bit trickier. Also, debugging operations should only be carried out on test or development servers as debugging can lock objects for long periods of time.

Debugging is a relatively straightforward process; however, if you have never used a debugger before it could take a while to get used to the tool and the techniques. If you have worked with a debugger in the past, then you should be able to begin using it immediately. Its features are very clear-cut for the experienced developer.

Intellisense in the Query Editor

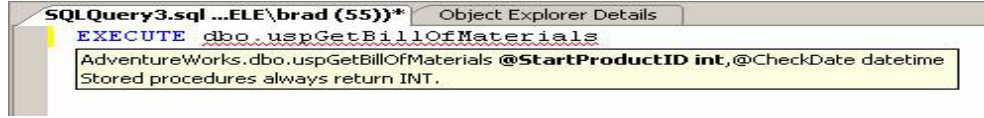
Some DBAs may remember the syntax of most T-SQL statements. For those of us less gifted, it means spending lots of time on Books Online looking up the structure and syntax of a particular statement to get it right. Any tools that we can take advantage of to write code faster, more accurately, and without referring to Books Online are winners in my book.

Intellisense, to the disappointment of many, was not included in the SSMS Query Editor in SQL Server 2005. It was one of the most requested features, and so Microsoft built it into SQL Server 2008. While this is a great addition to SQL Server, there are some limitations. One such limitation is that it only works with SQL Server 2008; it does not provide backwards compatibility with SQL Server 2005. Also, it does not possess refactoring or advanced formatting capabilities. Even with these limitations, Intellisense is an extremely valuable feature; syntax errors are very obvious, highlighted by the infamous red squiggly underline. This feature is very easy to use in the Query Editor; you simply need to start typing and Intellisense will take care of the rest. In addition to underlining incorrect syntax, Intellisense will automatically:

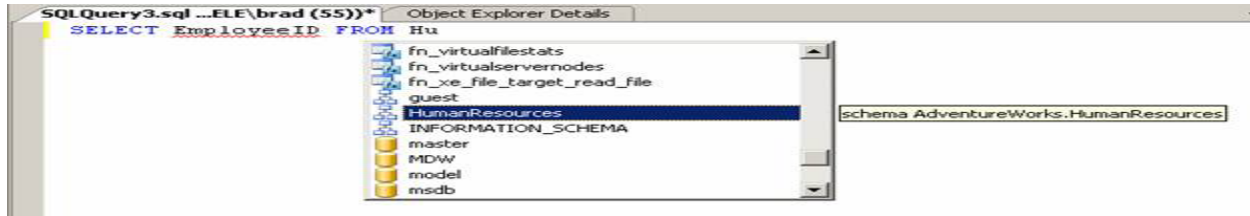
- Complete a word in a variable, command, or function as you type.



- List the available parameters required by a function or stored procedure.



- Open a list that provides available database objects and user-defined variables that you have previously declared.



Intellisense is a great new feature added to SQL Server 2008. If you have not used it before, it can take a little time to get comfortable with it. However, once you are used to it, writing T-SQL code becomes easier, quicker, and more intuitive than in the past.

Activity Monitor

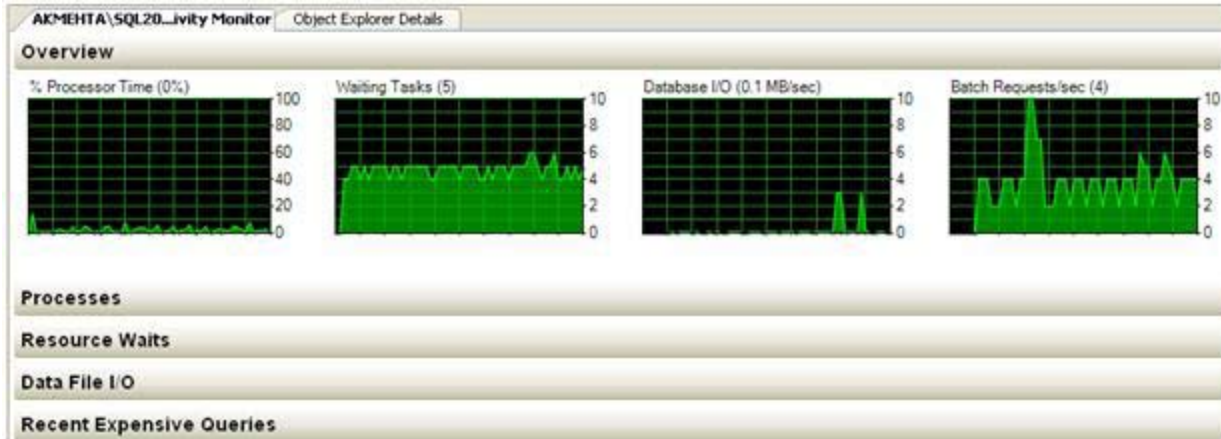
SQL Server 2005 provided an easy to find, easy to run Activity Monitor to view process information. This early version displayed information on Locks by Objects and Locks by Process. The Activity Monitor in SQL Server 2008 is a bit more challenging to find and use, but many great features have been added to give DBAs a quick view into SQL Server 2008 systems performance. The new Activity Monitor includes performance dashboards with graphs and performance indicators with filtering capabilities, making it easier for DBAs find important performance metrics.

To start Activity Monitor in SQL Server 2008, launch the SQL Server Management Studio. There are a couple different ways to launch the Activity Monitor. You can either click the Activity Monitor button on the standard toolbar or right-click the server instance in Object Explorer and click the Activity Monitor (see figure at right).



Right-Click Server Instance to Access Activity Monitor

Once you launch the Activity Monitor the following screen will appear:



Main Interface for the Activity Monitor

The first thing you will notice is that the Activity Monitor in SQL 2008 looks totally different from SQL 2005. There are 4 graphs displaying % Processor Time, Waiting Tasks, Database I/O, and Batch Requests/sec. In the past, if you needed a quick overview of these metrics, you would have to pull it from a separate tool such as the System Monitor.

After the Overview Section, the Activity Monitor contains four additional sections including; Processes, Resource Waits, Data File I/O, and Recent Expensive Queries (see Figure 3).

Processes Section – Click on and expand the Processes section (see Figure 4) to monitor the status of any SQL Server Process that is currently running on the system including: Session ID, User Process, Login Details, Database used by the process, Task Status, SQL Command run by the process, Application Used, Memory Used, Host Name, and more. The processes section also allows you to filter and sort using drop-down boxes at the top of each column.

Session ID	User Process	Login	Database	Task State	Command	Application	Wait Time (ms)	Wait Type	Wait Resou	Blk By	He Blz	Memory Use (KB)	Host Name	Wof Gros
51	1	akmehta				SQLAgent - Job invocation engine	0				1	16	AKMEHTA	
52	1	akmehta				SQLAgent - Generic Refresher	0				1	16	AKMEHTA	
53	1	akmehta				Microsoft SQL Server Management Studio	0				1	16	AKMEHTA	
54	1	akmehta				Microsoft SQL Server Management Studio	0				1	16	AKMEHTA	
55	1	akmehta	tempdb	RUNNING	SELECT	Microsoft SQL Server Management Studio	0				1	16	AKMEHTA	
56	1	akmehta				Microsoft SQL Server Management Studio	0				1	16	AKMEHTA	
57	1	akmehta				Microsoft SQL Server Management Studio	0				1	16	AKMEHTA	
58	1	akmehta				Microsoft SQL Server Management Studio	0				1	0	AKMEHTA	

Activity Monitor Processes

One really nice feature in the processes section is the ability to launch a SQL Profiler Trace for a specific SPID that is having issues. You simply select the SPID in question, right click, and choose “Trace Process in SQL Server Profiler” option.

Resource Waits – This section (see Figure 5) provides information on key resources waiting on the server. Specifically, “Resources Waits” measure the amount of time a SQL worker thread needs to wait until it can gain access to other resources on the server such as Memory or CPU. A high resource wait time could hint at a resource bottleneck that needs to be resolved. Resource wait metrics captured are Backup, Buffer I/O, Buffer Latch, CPU, Lock, Memory, Network I/O, etc., and you can sort and filter on any column.

Wait Category	Wait Time (ms/sec)	Recent Wait Time (ms/sec)	Average Waiter Count	Cumulative Wait Time (sec)
Signal	5208	4537	5.0	43022.4
Backup	0	28	0.0	11
Buffer I/O	0	10	0.0	158
Buffer Latch	0	0	0.0	0
CPU	0	0	0.0	3
Latch	0	0	0.0	0
Lock	0	0	0.0	23
Logging	0	0	0.0	14
Memory	0	0	0.0	0
Network I/O	0	0	0.0	23

Activity Monitor Resource Waits

Data File I/O – This section (see figure below) displays disk I/O information on system and user databases. If you suspect that a database is being affected by heavy disk I/O, you can find out using the Disk File I/O section. Information is displayed for each database on metrics like MB/Sec Read, MB/Sec Written, and Response Time, and can be sorted and filtered by column.

Database	File Name	MB/sec Read	MB/sec Written	Response Time (ms)
AdventureWorks	D:\Program Files\Microsoft SQL Server\MSSQ...	0.0	0.0	0
AdventureWorks	D:\Program Files\Microsoft SQL Server\MSSQ...	0.0	0.0	0
AdventureWorksDw2008	D:\Program Files\Microsoft SQL Server\MSSQ...	0.0	0.0	0
AdventureWorksDw2008	D:\Program Files\Microsoft SQL Server\MSSQ...	0.0	0.0	0
AdventureWorksLT2008	D:\Program Files\Microsoft SQL Server\MSSQ...	0.0	0.0	0
AdventureWorksLT2008	D:\Program Files\Microsoft SQL Server\MSSQ...	0.0	0.0	0
master	D:\Program Files\Microsoft SQL Server\MSSQ...	0.0	0.0	0
master	D:\Program Files\Microsoft SQL Server\MSSQ...	0.0	0.0	0
model	D:\Program Files\Microsoft SQL Server\MSSQ...	0.0	0.0	0
model	D:\Program Files\Microsoft SQL Server\MSSQ...	0.0	0.0	0

Activity Monitor Data File I/O

Recent Expensive Queries – This section (see figure below) helps DBAs identify performance issues related to resource-intensive queries by Server Instance. This screen will display the most recent expensive queries executing against the server. You can sort or filter queries by column to help identify the problem queries. Then, if you right click the query and choose the “Edit Query Text” option, SQL Server allows you to view the entire query. You can also view a graphical execution plan of the query by choosing the “Show Execution Plan” option.

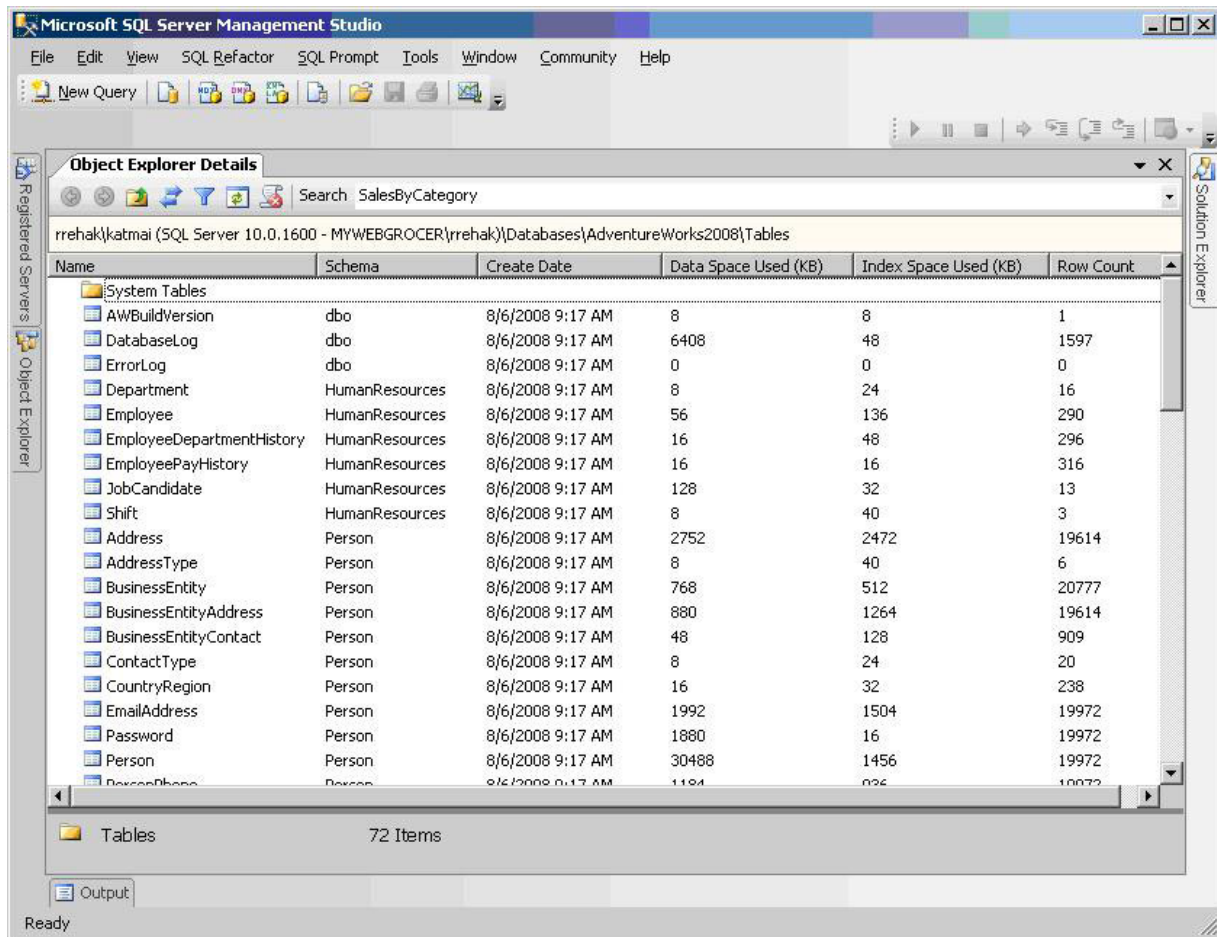
Query	Executions/m	CPU (ms/sec)	Physical Reads/sec	Logical Writes/sec	Logical Reads/sec	Average Duration (ms)	Plan Count
SELECT COUNT (DISTINCT c2) FROM Table...	480	45796	353068	7	1068582	124444	1
SELECT 'Total income is', ((OrderQty * UnitPric...	240	5107	426	0	7869	39380	1
SELECT p.Name AS ProductName, i.NonDisc...	180	4275	387	0	6078	2645	1
SELECT p.[Name], AVG (pch.StandardCost) A...	49200	3204	0	0	23780	4	1
SELECT ProductID, OrderQty, LineTotal FROM...	180	2973	0	0	3738	1259	1
SELECT p1.ProductModelID FROM Productio...	26640	1213	0	0	583865	3	1
SELECT ProductID, LineTotal FROM Sales.Sa...	360	917	0	0	7476	164	2
SELECT SalesOrderID, SUM(LineTotal) AS Su...	180	857	0	0	3738	295	1
SELECT ProductID, OrderQty, UnitPrice, LineT...	180	685	0	0	3738	321	1

Activity Monitor Recent Expensive Queries

Object Explorer Details

In SQL Server 2005, the Object Explorer “details” screen added no additional information than what already appeared in the Object Explorer itself. The details section has changed in SQL Server 2008. The details section displays a variety of useful information about the object selected in Object Explorer. To access the details section as you browse objects, you can press F7 or select “Object Explorer Details” from the “View” menu up above. There are a number of default columns, but you can also add additional columns by right clicking the column headings and selecting additional columns. The details window will display different information, depending on the type of object chosen (a database, a table, etc.). The following snapshot (see Figure 8) shows detail information about the tables in the AdventureWorks database, including information on schema, data space used, index space used, and total number of rows.

Database objects will display 5 columns of data by default, but you can select up to 36 columns for each database. Thirty-six columns is a lot of information to access inside the SSMS, so one solution is to select the rows, copy the information to the clipboard (CTRL-C) and then paste it directly into Excel workbook for easy viewing. Also, once you add additional columns, SQL Server is smart enough to remember those changes the next time you access the object details. The Object Explorer details window is a great way to return quick information on the status of your databases.



Object Explorer Details on Tables

Object Search

Another great new feature in the Object Explorer Details window is object search. This toolbar (see figure above) allows you to type in the name of an object and perform a search that is based on your current context. For example, to search for a table in a single database, you need to select the database in the Explorer window and then perform your search. To search in all of the databases, you would select the Databases node. Object search is limited to a single database or all databases, nothing in between. Once a search is performed, a list of objects, including the path to the object, will be displayed. Once you find your object, you can right-click, choose the "Synchronize" option, and SSMS will navigate to the object opening and expanding the parent nodes in the Explorer window. In SQL Server 2005, searching and finding objects by name was not an easy task. With SQL Server 2008, searching for objects in one or all databases using Object Search is a quicker, faster, and more intuitive process.

Multi-Server Queries

In the past, DBAs used third-party tools to query multiple servers at the same time. An exciting new feature in SQL 2008 is the ability to query multiple servers simultaneously and return the results to a single window in SSMS.

It is easy to set up. First, you will need to create a Server Group in the "Registered Servers" window and add specific SQL Server instances to that group. Then right-click the Server Group and select "New Query" and the Query Editor window will open. Finally, enter your T-SQL code, click "Execute," and the query will run on all the registered servers in the Server group.

Conclusion

Some great new features were added to the SQL Server Management Studio in SQL Server 2008. Microsoft has succeeded in morphing the SSMS in SQL Server 2008 into a much more useful tool kit for the DBA making routine DBA tasks a little easier and more intuitive to use. This white paper sampled a few of the latest and greatest features and is a good jumping-off point to get your feet wet with the SSMS in SQL Server 2008.

Learn More

Learn more about how you can improve productivity, enhance efficiency, and sharpen your competitive edge. Check out the following Global Knowledge courses:

[Implementing and Maintaining Microsoft SQL Server 2008 Analysis Services](#)

[Implementing and Maintaining SQL Server 2008](#)

[Implementing and Maintaining SQL Server 2008 Analysis Services and Business Intelligence Solutions](#)

[Maintaining, Troubleshooting, and Developing Solutions with Microsoft SQL Server 2008 Reporting Services](#)

[Writing Queries Using Microsoft SQL Server 2008 Transact-SQL](#)

For more information or to register, visit www.globalknowledge.com or call **1-800-COURSES** to speak with a sales representative.

Our courses and enhanced, hands-on labs offer practical skills and tips that you can immediately put to use. Our expert instructors draw upon their experiences to help you understand key concepts and how to apply them to your specific work situation. Choose from our more than 700 courses, delivered through Classrooms, e-Learning, and On-site sessions, to meet your IT and management training needs.

About the Author

Michael Manning is a Global Knowledge instructor and consultant specializing in Microsoft Sharepoint and SQL Server technologies. He has been working with SQL Server going back to the SQL 7.0 days and currently resides in Seattle, Washington.