

---

# SQL Server 2008 Metro Resource Governor Hands-on Lab

---

---

# Table of Contents

---

Lab Introduction.....	3
Objectives .....	3
Prerequisites.....	3
Lab Scenarios .....	3
Virtual PC Configuration and Setup .....	4
Using Virtual Machines .....	4
Tips on how to successfully complete the lab exercises.....	6
Resource Governor in Action .....	7
Exercise 1: Configuring Resource Governor .....	8
Exercise 2: Monitoring Resource Usage under the Resource Governor .....	10

# Lab Introduction

---

## Objectives

The goal of these hands-on lab materials is to get an understanding of when to use one of the more advanced features of SQL Server 2008: Resource Governor.

The intent of these exercises is to provide you with best practices and implementation details for SQL Server 2008. As with all software development projects, your production environment may differ from this build and this environment. Be sure to design, implement, and test your final architecture extensively to minimize downtime and data loss.

For the latest details on SQL Server 2008, please visit <http://www.microsoft.com/sql/2008/>.

## Prerequisites

After completing these self-paced labs, you will be able to:

- Understand appropriate uses for Resource Governor
- Create Resource Pools
- Create Workload Groups
- Monitor Resource Usage

- Experience with SQL Server Administration
- Experience with the SQL Server 2005 Tools or SQL Server 2008 Tools is very helpful but not required
- Familiarity with the Transact-SQL language is very helpful but not required
- Desire to sink your teeth into SQL Server 2008!

## Lab Scenarios

### 1. **Resource Governor in Action**

This series of exercises is meant to show you how Resource Governor works and how it can be used to limit workload resource usage when a higher priority workload starts running concurrently.

**Time for Completion:** 30-45 minutes

## Virtual PC Configuration and Setup

This Virtual PC environment contains five already installed instances of SQL Server running on Windows 2003 Server SP2. The VPC machine name is **CHICAGO**.

The instances, versions and environments are described in the following table:

SQL Server Service Name	SQL Server Version Edition
MSSQL\$SQLDEV01	SQL Server 2008 build 10.00.1300.13 (February 2008 CTP) Developer Edition
MSSQL\$SQLDEV02	SQL Server 2008 build 10.00.1300.13 (February 2008 CTP) Developer Edition
MSSQL\$SQLEXPRESS	SQL Server 2005 build 9.00.3054.00 (SP2) Express Edition with Advanced Services
MSSQL\$SQLDEV03	SQL Server 2008 build 10.00.1300.13 (February 2008 CTP) Developer Edition
MSSQL\$CONFIGSERVER	SQL Server 2008 build 10.00.1300.13 (February 2008 CTP) Developer Edition

The accounts and passwords used in the following exercises are shown in the following table:

Account Name	Account Password	Account Usage
Administrator	Pass@word1	Login account for VPC
SQLService	Pass@word1	Startup account for all SQL Server services
sa	Pass@word1	SQL Server System Administrator

### Using Virtual Machines

Using Virtual Machines allows you to isolate and control an environment independently from your host operating system. Your host operating system can run any operating system that supports Virtual PC or Virtual Server. Then, within your Virtual Machine, you can run any "base" operating system desired. In the case of this self-paced lab series, the base operating system is Windows Server 2003 Enterprise Edition with Service Pack 2. A virtual machine has multiple components: a virtual hard drive (\*.vhd) and a virtual machine configuration file (\*.vmc). With Virtual PC or Virtual Server, you can open the vmc file in order to configure your virtual machine. The initial configuration of this vmc is that it expects 1GB of memory allocated to the virtual machine; this is memory in addition to what your host operating system needs. If you do not have more than 1.5GB of physical memory, reconfigure the virtual machine to use 512MB of memory instead.

Once setup for your environment, start (or turn on) your virtual machine. Once started, you will see a login dialog. In order to access the login dialog within your virtual machine, there are special keystrokes required. The details for the login are provided in the following table:

Tasks	Detailed Steps
Login to your VPC image.	To access the login dialog on the VPC, use Right-Alt + Del and then login in as <b>Administrator</b> with a password of <b>Pass@word1</b>

Within Virtual PC or Virtual Server, most commands focus on using the **Right-alt** key. The **Right-alt** is the alt key immediately to the right of the spacebar.

Command Description	Keystroke Combination
Login	<b>Right-alt</b> + delete
Full Screen/Return to Host (in VPC)	<b>Right-alt</b> + enter

### Customize the page file

The page file size has been reduced to 0 bytes in order to minimize the cost of expanding the .vhd (when you ran setup to install this VPC image). If you do not have 1.5-2GB of **physical** memory on your machine, you will need a page file. You have two options:

- Set the Page File to System Managed Size (**recommended**)
- Increase the size of the "custom" configuration

If desired, use the steps below to configure your Page File to a System Managed Size.

Tasks	Detailed Steps
Customize your Page File settings.	<ol style="list-style-type: none"> <li>1. Start the <b>Control Panel, System</b> applet. From the Windows task bar, select the <b>Start   Control Panel   System</b> menu item.</li> <li>2. In the System Properties dialog, select the Advanced Tab.</li> <li>3. In the <b>Performance</b> section, click the <b>S</b>ettings button.</li> <li>4. In the Performance Options dialog, select the Advanced Tab.</li> <li>5. In the <b>Virtual Memory</b> section, click the <b>C</b>hange button.</li> <li>6. Select the <b>System managed size</b> radio button in the Virtual Memory dialog. This radio button is in the <b>Paging file size for select drive</b> section. NOTE: Drive C: will be selected as it is the only drive available to our VPC.</li> <li>7. Select <b>S</b>et.</li> <li>8. Click <b>OK</b> and continue to click OK as you exit the Control Panel, System dialogs.</li> </ol>

## Tips on how to successfully complete the lab exercises

### **IMPORTANT**

For best results, read the Tasks column and its associated notes first. Then proceed to follow the detailed steps for step by step instructions on what to verify, review, execute, etc. Please check off steps as you complete them and execute steps in the order in which they are listed. As a best practice, read the entire step, think about what it is meant to accomplish and then execute it.

There are numerous steps that you do NOT immediately execute and instead may have you review and/or alter what you might expect your normal behavior to be in order to show you a new feature or two. There are lots of notes along the way, all of which give you insight into special features, cool tips and often best practices. It's not a speed contest to finish first. Take it slow, review the steps and you'll get a lot more out of your workshop experience.

Finally, as the complexity of the exercise increases, the need to follow steps in a certain order also increases. Some exercises may fail if steps are missed and worse yet, other exercises might require you to start over if/when steps are missed.

**Consider using a pen/pencil to mark off steps as you complete them. Additionally, consider taking a step back from each step to think about what you're trying to accomplish. This will help to minimize errors and increase learning as you may be able to predict certain steps and/or think of additional things to test and learn!**

---

# Resource Governor in Action

---

## Scenario


In this scenario, you will setup Resource Governor to allow workloads from the Development and Marketing teams in your company to run concurrently on your SQL Server 2008 instance. You want to ensure that the Development team gets adequate resources when running a workload and you know that the Resource Governor feature of SQL Server 2008 will allow you to do that.

The first exercise deals with setting up Resource Governor for the scenario above. In the second exercise, we'll introduce the two workloads and monitor the resource usage to watch the workload throttling occur.

The required resources needed for this lab are already started automatically when the VPC starts.

## Exercise 1: Configuring Resource Governor

In this lab exercise, the intent is not to learn every line of code and/or every feature, but instead, learn the basics of how Resource Governor works. In this case, we will setup two Resource Pools and Workload Groups to allow connections to be classified as either Development or Marketing and bound to the correct Resource Pool and Workload Group.

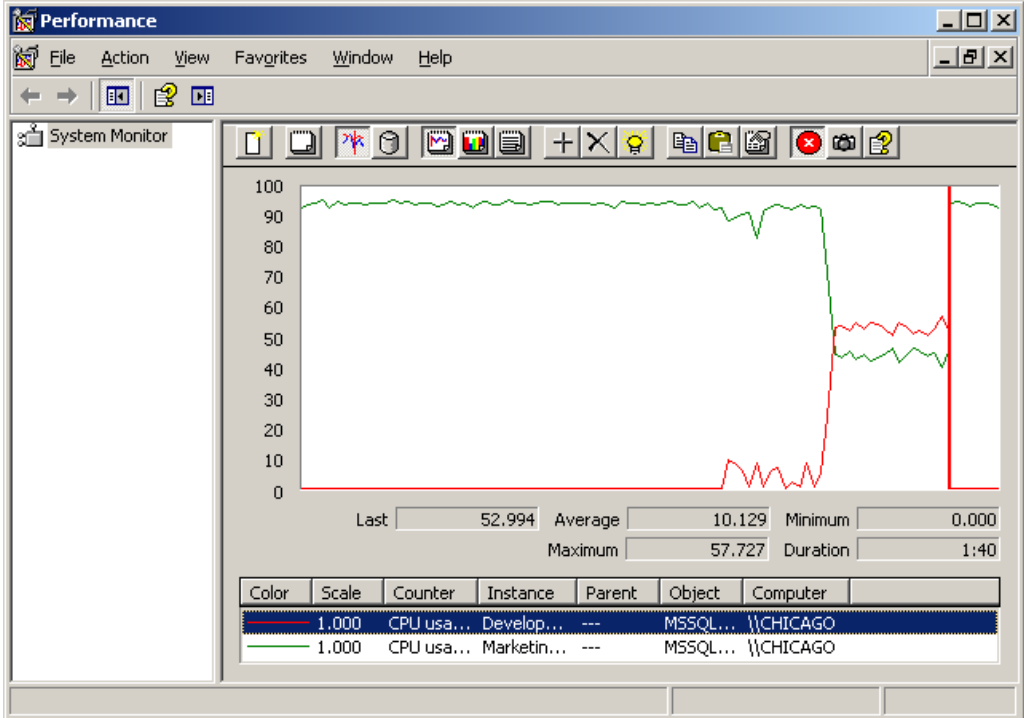
Tasks	Detailed Steps
Setup Resource Governor	<ol style="list-style-type: none"> <li>1. If open, close <b>SQL Server Management Studio</b>.</li> <li>2. Start <b>My Computer</b> From the Windows task bar, select <b>Start   My Computer</b> <i>OR</i> Press <i>WindowsKey+E</i> The <i>WindowsKey</i> does not exist on every keyboard but when it does, it resides between the Ctrl and the Alt keys, sometimes next to a Fn key. The icon on the key is usually the Windows logo  .</li> <li>3. Navigate to the following directory:  <b>C:\Metro Demos\Troubleshooting Demos</b></li> <li>4. Double-click <b>Troubleshooting Demos.ssmss1n</b>.</li> <li>5. By opening a solution file (<b>SQL Server Management Studio Solution</b>), you can have a group of scripts all centralized for ease of use and management. Once within <b>SQL Server Management Studio</b>, navigate to the <b>Solution Explorer</b> window. If this window does not appear, you can add this window back into SQL Server Management Studio by using the <b>View, Solution Explorer</b> menu item.</li> <li>6. From the list of projects within the Solution <b>Troubleshooting Demos</b>, find the <b>Queries</b> project. Double-click <b>SetupResourceGovernor.sql</b>.</li> <li>7. Review and execute lines 24-29 to query the current state of Resource Governor. You should see that the <code>classifier_function_id</code> is 0. This means that Resource Governor has not been setup.</li> <li>8. Review and execute lines 33-39. These create two Resource Pools – one each for the Marketing and Development groups. Notice the <code>MAX_CPU_PERCENT</code> specified for each. The idea is that in a situation where both workloads are running and competing for CPU resources, the Marketing workload would be limited to no more than 10% of the available CPU resources. At the same time, the Development group is limited to 90%, so that the Marketing group has its 10%.</li> </ol> <p>Note: The various maximum CPU percentages for the defined Resource Pools must not exceed 100%. If the combined total does not equal 100% (for instance if we had defined two Resource Pools to have a maximum of 40% each), then the remaining available CPU resources are shared between the Resource Pools.</p> <p>Resource Pools can have limits defined for CPU and memory, but not for IO, and there can be minimum and maximum limits. Resource Governor does not support IO limits.</p> <p>There are many options available when defining Resource Pools to configure the various limits and behaviors. Please look in the SQL Server 2008 Books Online entry for <code>CREATE RESOURCE POOL</code> for more details.</p>

Tasks	Detailed Steps
	<p>9. Review and execute lines 42-43 to view the defined Resource Pools. Notice that the two Resource Pools we just defined do not appear. This is because any changes to the Resource Governor require it to be reconfigured before they are active. Further notice that there are two Resource Pools we did not define – <i>internal</i> and <i>default</i>.</p> <p>Note: The <i>internal</i> Resource Pool is system-defined and is for all background tasks within SQL Server (such as the ghost-cleanup task, or the lazywriter task). This Resource Pool has no limits and if necessary will pressure other Resource Pools for CPU or memory if needed, overriding whatever their settings are.</p> <p>The <i>default</i> pool is where connections are put that are not specifically bound to a user-defined Resource Pool. Again, it is system-defined and has no limits. However, this Resource Pool will not pressure other Resource Pools – in fact, it will be pressured by all other Resource Pools as it is the least ‘important’ Resource Pool in the instance.</p> <p>10. Review and execute lines 46-49 to reconfigure the Resource Governor and view the defined Resource Pools again. This time our new Resource Pools are listed as expected.</p> <p>11. Review and execute lines 52-58 to define two Workload Groups, one each for the Marketing and Development teams. Each Resource Group is bound to its respective Resource Pool. Note: There are many options available when defining Workload Groups to configure the various limits and behaviors. Please look in the SQL Server 2008 Books Online entry for <code>CREATE WORKLOAD GROUP</code> for more details.</p> <p>12. Review and execute lines 61-62 to view the defined Workload Groups. Again, we see that our two new Workload Groups do not appear because the Resource Governor has not yet been reconfigured. Further notice that there are two Workload Groups we did not define – <i>internal</i> and <i>default</i>. These are the matching system-defined Workload Groups to the system-defined Resource Pools.</p> <p>13. Review and execute lines 65-68 to reconfigure the Resource Governor and check that our two new Workload Groups are now listed.</p> <p>14. Review and execute lines 73-76 to create two dummy databases, <b>MarketingDB</b> and <b>DevelopmentDB</b>, that we will use to simulate workloads from the Marketing and Development teams, respectively.</p> <p>15. Review and execute lines 79-95 to define the Classifier Function. Note that it analyzes the <code>ORIGINAL_DB_NAME</code> of the server connection and returns the name of a Workload Group, depending on the database name.</p> <p>When a connection is made to a SQL Server 2008 instance running Resource Governor, it must be classified into one of the defined Workload Groups so that any queries run on that connection are properly governed. This is done by the Classifier Function. The sole purpose of the Classifier Function is to output the name of a Workload Group and it can have use any T-SQL logic and available information to do that. In this case, we are basing the Workload Group on the database that the connection was made to, but you can imagine a function that uses the time of day, user names, or client names to make decisions, for instance.</p> <p>16. Review and execute lines 98-105 to register the Classifier Function, reconfigure the Resource Governor, and view its state. You will notice that the <code>classifier_function_id</code> is now non-zero. Resource Governor is running!</p>

## Exercise 2: Monitoring Resource Usage under the Resource Governor

In this exercise, you will see the impact of the Resource Governor on a system with concurrent workloads competing for CPU resources.

Tasks	Detailed Steps
<p>Use the System Monitor to view CPU usage in each of the Resource Pools defined above.</p>	<p>In this exercise, you will start the System Monitor and view the CPU usage of multiple workloads and how they are affected by the Resource Governor.</p> <ol style="list-style-type: none"> <li>1. Start the System Monitor. From the Windows task bar, select <b>Start   Administrative Tools   Performance</b>.</li> <li>2. Select <b>A</b>ction   New <b>W</b>indow From Here.</li> <li>3. Open the <b>Add Counters</b> dialog by pressing the icon with the + symbol, or pressing CTRL+I.</li> <li>4. In the <b>Performance object:</b> drop-down, select <b>MSSQL\$SQLDEV01:Resource</b>.</li> <li>5. In the <b>Select counters from list:</b> scrolling-menu, select <b>CPU usage %</b>.</li> <li>6. In the <b>Select instances from list:</b> box, select <b>DevelopmentPool</b> and <b>MarketingPool</b>. (Select <b>DevelopmentPool</b> first, the press-and-hold the CTRL key and select <b>MarketingPool</b>. Release the CTRL key and they should both be selected.)</li> <li>7. Select <b>A</b>dd and then <b>C</b>lose.</li> <li>8. The performance trace will start and you will see that both Resource Pools have zero CPU usage.</li> <li>9. Start a <b>Command Prompt</b> window by clicking the Command Prompt icon in the Quick Start section of the Task Bar.</li> <li>10. Enter the following command in the <b>Command Prompt</b> window: <pre>cd C:\Metro Demos\Troubleshooting Demos\Troubleshooting Demos</pre> </li> <li>11. We will now start a workload using the <b>MarketingDB</b> database. The workload will simply create an index and repeatedly run <code>ALTER INDEX ... REORGANIZE</code> on it. This will use CPU resources without using IO resources, allowing us to experiment with pure CPU workloads.</li> <li>12. Enter the following command in the <b>Command Prompt</b> window: <pre>sqlcmd /E /S.\SQLDEV01 /dMarketingDB /iRunQueries.sql</pre> </li> <li>13. Go to the <b>Performance</b> window and notice that the CPU usage for the Marketing Resource Pool has risen to almost 100%. (The exact amount will vary depending on the machine hosting the VPC.) <p>How can this be? The connection specified the <b>MarketingDB</b> database, and should have been classified into the <b>MarketingGroup</b> Workload Group, linked to the <b>MarketingPool</b> Resource Pool with a 10% maximum CPU limit. In fact it was, but this behavior illustrates an important point of Resource Governor – workload throttling only occurs when it HAS to. In this case, there are no other workloads competing for resources so the MarketingPool is permitted to use as much CPU as it needs.</p> </li> </ol>

Tasks	Detailed Steps																					
	<p>14. Now start another <b>Command Prompt</b> window.</p> <p>15. Enter the following command in the <b>Command Prompt</b> window:</p> <pre>cd C:\Metro Demos\Troubleshooting Demos\Troubleshooting Demos</pre> <p>16. We will now start a workload using the <b>DevelopmentDB</b> database. The workload will simply create an index and repeatedly run <code>ALTER INDEX ... REORGANIZE</code> on it. This will use CPU resources without using IO resources, allowing us to experiment with pure CPU workloads – it does the same as the workload we started on the <b>MarketingDB</b> database.</p> <p>17. Enter the following command in the <b>Command Prompt</b> window:</p> <pre>sqlcmd /E /S.\SQLDEV01 /dDevelopmentDB /iRunQueries.sql</pre> <p>18. Go to the <b>Performance</b> window and notice that the CPU usage for the MarketingPool Resource Pool drops down as the DevelopmentPool CPU usage ramps up.</p> <p>You should see something like below.</p>  <p>The screenshot shows the Performance Monitor window with a line graph of CPU usage over time. The y-axis represents CPU usage percentage from 0 to 100. Two data series are plotted: a red line for 'DevelopmentDB' and a green line for 'MarketingDB'. The MarketingDB usage starts at approximately 95% and remains high until a vertical red line indicates the start of the DevelopmentDB workload. At this point, the MarketingDB usage drops sharply to about 45%, while the DevelopmentDB usage rises to about 55%. Below the graph, a table lists the performance counters for both instances.</p> <table border="1" data-bbox="673 1486 1425 1570"> <thead> <tr> <th>Color</th> <th>Scale</th> <th>Counter</th> <th>Instance</th> <th>Parent</th> <th>Object</th> <th>Computer</th> </tr> </thead> <tbody> <tr> <td>Red</td> <td>1.000</td> <td>CPU usa...</td> <td>Develop...</td> <td>---</td> <td>MSSQL...</td> <td>\\CHICAGO</td> </tr> <tr> <td>Green</td> <td>1.000</td> <td>CPU usa...</td> <td>Marketin...</td> <td>---</td> <td>MSSQL...</td> <td>\\CHICAGO</td> </tr> </tbody> </table> <p>19. It's clear that the <b>MarketingDB</b> workload was throttled to allow the <b>DevelopmentDB</b> workload to get the CPU it needed. Resource Governor in action!</p> <p>20. In <b>SQL Server Management Studio</b> select <b>File, Exit</b>. If query windows are open, do NOT save when exiting. Shutdown your VPC and do NOT save any changes..</p>	Color	Scale	Counter	Instance	Parent	Object	Computer	Red	1.000	CPU usa...	Develop...	---	MSSQL...	\\CHICAGO	Green	1.000	CPU usa...	Marketin...	---	MSSQL...	\\CHICAGO
Color	Scale	Counter	Instance	Parent	Object	Computer																
Red	1.000	CPU usa...	Develop...	---	MSSQL...	\\CHICAGO																
Green	1.000	CPU usa...	Marketin...	---	MSSQL...	\\CHICAGO																

