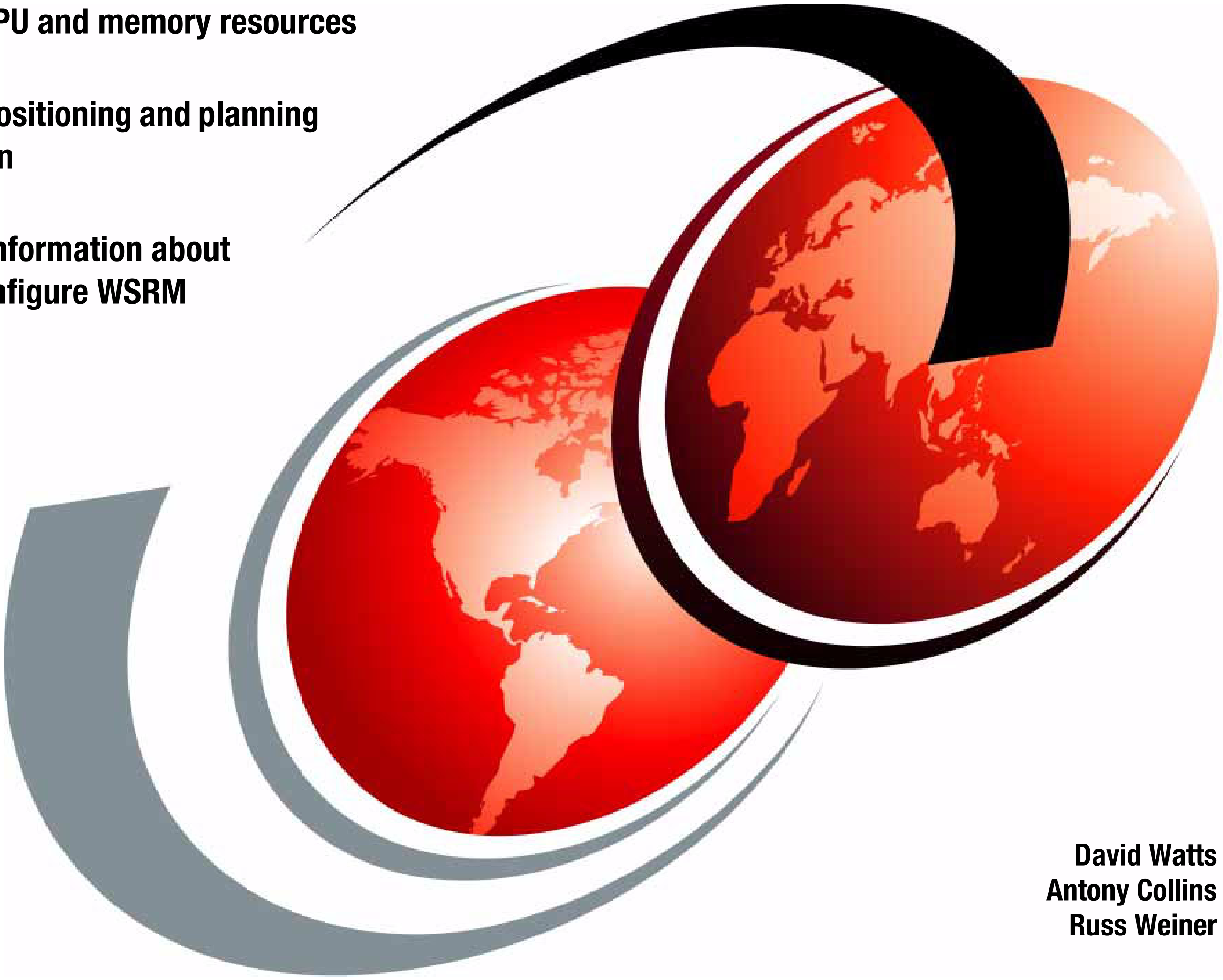
Implementing Windows System Resource Manager

Describes how WSRM helps you manage CPU and memory resources

Includes positioning and planning information

Provides information about how to configure WSRM



David Watts
Antony Collins
Russ Weiner

Redpaper

ibm.com/redbooks

International Technical Support Organization

Implementing Windows System Resource Manager

October 2003

Note: Before using this information and the product it supports, read the information in “Notices” on page v.

First Edition (October 2003)

This edition applies to Microsoft Windows System Resource Manager, part of Windows Server 2003, Datacenter Edition.

Contents

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:
This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

DB2®
eserver™
eserver ™
IBM®
ibm.com®
Redbooks™
Redbooks(logo) ™
Tivoli®
xSeries®

The following terms are trademarks of other companies:

Intel, Intel Inside (logos), MMX, and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.

Preface

This Redpaper introduces Microsoft® Windows System Resource Manager (WSRM), a component of Windows Server 2003, Datacenter Edition and Enterprise Edition. WSRM is a utility used for managing CPU and memory resource allocation among multiple applications that run on the same operating system image.

This paper describes what the tool does and how it works. It also describes how to prepare to use WSRM and how to implement a basic configuration of WSRM.

The paper will help Windows Server 2003 customers, especially those implementing an IBM @server xSeries® 445 solution, and those people who want to understand how CPU and memory resource usage can be managed.

The team that wrote this Redpaper

This Redpaper was produced by a team of specialists from around the world working at the International Technical Support Organization, Raleigh Center.

David Watts is a Consulting IT Specialist at the International Technical Support Organization in Raleigh. He manages residencies and produces IBM® Redbooks™ on hardware and software topics related to IBM @server xSeries systems and associated client platforms. He has authored over 30 Redbooks and Redpapers; his most recent include *Implementing Systems Management Solutions using IBM Director* and the *IBM @server xSeries 445 Solution Assurance Product Review Guide*. He has a Bachelor of Engineering degree from the University of Queensland (Australia) and has worked for IBM for more than 15 years. He is an IBM @server Certified Specialist for xSeries and an IBM Certified IT Specialist.

Antony Collins is an xSeries Technical Specialist in Sydney, Australia. He has seven years of experience in Windows® systems and operational management. His areas of expertise include solution design, systems management, and xSeries server architecture. As an xSeries Specialist for IBM Australia, Antony provides pre-sales xSeries support and design assistance to IBM professionals, IBM Business Partners, and IBM customers. Antony is a Certified Specialist for xSeries, Microsoft Windows NT® and 2000 MCP, and also a Certified Novell Administrator (CNA).

Russ Weiner is an xSeries Technical Support Specialist in Seattle, Washington. He has 20 years of experience in information technology as a programmer, network administrator, in mainframe operations, helpdesk, and data security. He has worked at IBM in the Advanced Technical Support organization for five years. His current area of focus is field technical support for xSeries servers.

The Redpaper team (left to right): Antony, David, and Russ

Thanks to the following IBM employees for their contributions to this project:

Donn Bullock, Global Brand Manager, xSeries 445, Raleigh
Michael L. Nelson, eServer Solutions Engineering, Raleigh
Michael S. Lee, IBM Center for Microsoft Technologies, Seattle
Bob Zuber, WW xSeries Product Marketing, Raleigh

Thanks also to the following Microsoft employees:

Kalyan Basu
Eileen Brown
Giovanni Marchetti
Lars Mikaelsson
Ji-Young Kim
John Knie
Aruna Koka
Thomas Kuklovszky
Andrew Page
Bernd Pfann
Vij Vasu

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll team with IBM technical professionals, Business Partners and/or customers.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you'll develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our papers to be as helpful as possible. Send us your comments about this Redpaper or other Redbooks in one of the following ways:

- Use the online **Contact us** review redbook form found at:

  ibm.com/redbooks

- Send your comments in an Internet note to:

  redbook@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HZ8 Building 662
  P.O. Box 12195
  Research Triangle Park, NC 27709-2195

1

Introduction

Microsoft Windows System Resource Manager (WSRM) is a utility used for managing CPU and memory resource allocation among multiple applications that run on the same Windows Server 2003 operating system image. It is a key factor in Microsoft's server consolidation strategy. The server component is available for use on:

- Microsoft Windows Server 2003, Enterprise Edition
- Microsoft Windows Server 2003, Datacenter Edition

WSRM is supplied on a separate CD as part of the operating system package and can be installed by inserting the CD and following the instructions. After WSRM is installed, it is started by clicking **Start** → **Programs** → **Administrative Tools** → **Windows System Resource Manager**.

> **Tip:** If you are using an evaluation version of Windows Server 2003, WSRM is not supplied on CD. The WSRM utility can be downloaded, in ISO (CD image) format, from:
>
> `http://www.microsoft.com/windowsserver2003/downloads/wsrm.mspx`

The client component runs on Windows XP, the Windows Server 2003 family, and Windows 2000 with Service Pack 2 or later.

This chapter introduces Windows System Resource Manager (WSRM) and the reasons why a customer would want to use this application. This includes its primary features, which provide the resource management that most customers lack, and how WSRM guarantees resource allocation to several applications coexisting on the same system.

This chapter includes the following sections:

- What WSRM does
- How WSRM works
- WSRM features

> **Note:** We frequently use two terms in this Redpaper. A *process* is an object created by Windows to represent an executing program. A *processor* is the CPU of the server.

1.1 What WSRM does

Typically, customers run multiple production applications on one server. These applications often compete for system resources, such as processor and memory. When competition is highest among applications (during peak periods, for example), the applications that need, for business reasons, to have the most resources available to them often don't get this allocation and performance suffers.

Older applications might also have the tendency not to release the resources they have been using after they have finished completing their tasks. Applications causing memory leaks also present significant problems to the function of the system, and to other applications that are installed.

> **Tip**: A memory leak is the gradual loss of available system memory when an application repeatedly fails to return the memory that it has obtained for use. As a result, the available system memory becomes exhausted, causing programs or applications to no longer function efficiently. Memory leaks are generally caused by bugs in the software and can lead to catastrophic effects on overall system performance.

All of these circumstances can cause the application or the operating system to become unstable. WSRM can help avoid these situations by specifically allocating and managing resources to applications, processes, and services, thereby restricting the applications from consuming more resources than they have otherwise been allocated.

1.2 How WSRM works

When an application starts, a *process* is created (normally named based on the name of the application's executable). The purpose of this process is to provide the necessary resources (virtual memory, executable code, threads, and so forth) for the application to function. When the process starts, the code that gets executed is run in streams of instructions known as threads. Each application or program has at least one process, and each process must have at least one, but typically has several threads (that is, the applications are multithreaded). So, in summary, processors execute threads belonging to processes, not applications.

The priority in which a thread is executed by the processors (in relation to all other threads) is called *thread priority.* Thread priority is partially determined by its parent process. WSRM can change the priority of the following four priority levels:

- Low
- BelowNormal
- Normal
- AboveNormal

WSRM cannot set "High" and "Realtime" priorities.

The priority of a running process can be seen through Windows Task Manager by right-clicking the process and selecting **Set Priority**. You can also see it in the Process tab of Task Manager. Click **View** → **Select Columns** and select **Base Priority**.

> **Tip:** Never manually change the priority of a process or application to "Realtime." Doing so provides almost all CPU time to that process, thereby reducing the other processes from having enough. This will result in system instability.

Normally, a process inherits its priority from its parent, that is, the process that started it. For example, if a process starts from within Windows Explorer (the user interface), it is assigned the “Normal” priority, because that is the priority of Windows Explorer. After launch, an application might alter the priority of its process, if it is coded that way.

WSRM is able to alter dynamically the priority class of processes running in Windows Server 2003. This also changes the priority of the threads that are being executed within the processes. By doing this, WSRM can provide varying access to system resources based on a *resource allocation policy*. In addition to managing the CPU usage of a process, WSRM also allows the controlling of the working set size of an application.

To determine what policy should be used to manage a process, WSRM uses *process matching criteria* to identify the process. Process matching criteria are name-based criteria used to group one or more processes for subsequent resource allocation or accounting. Each process to be managed must be defined using this criteria. When a process executes, WSRM matches that process using the process matching criteria. After a successful match has been made, WSRM then governs CPU and memory allocations for that process.

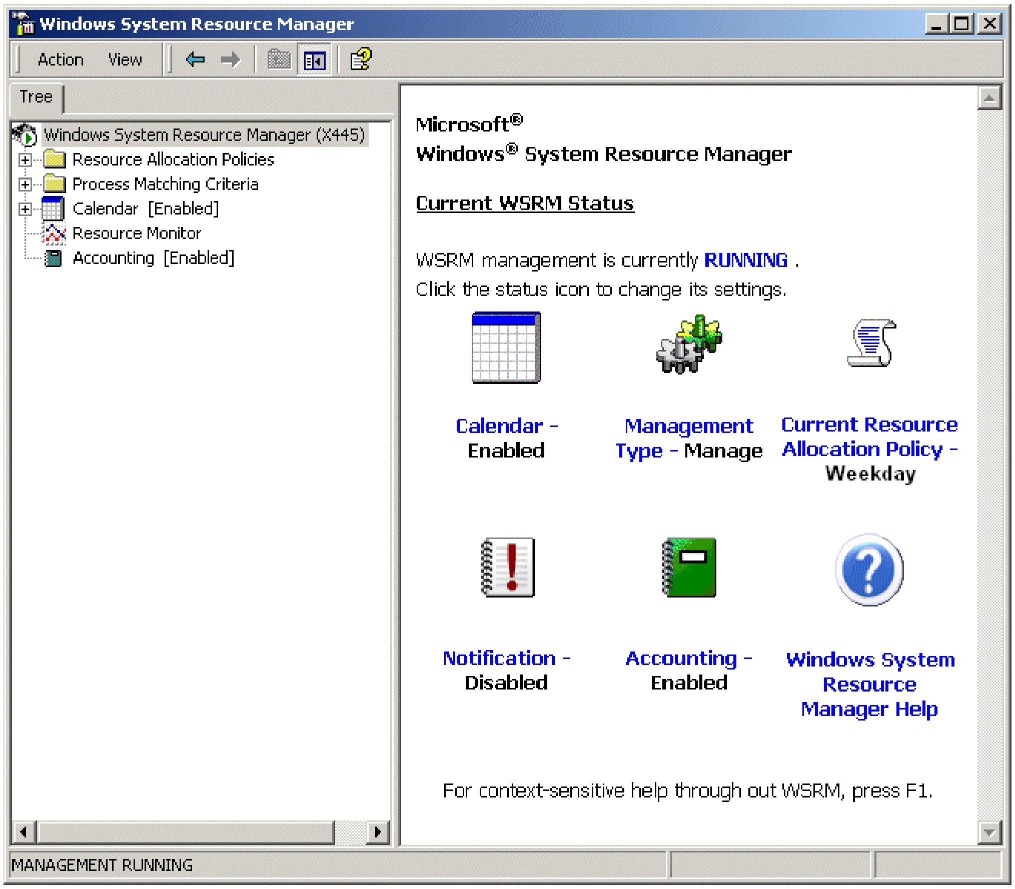
There are two types of policy types within WSRM:

- Managing policy — The resource allocation policy currently set to manage the server. A managing policy is the policy that is currently active and, therefore, managing the resources on the system. There can only be one managing policy active at any one time.
- Profiling policy — A profiling policy enables administrators to see which processes match the process matching criteria without actually applying its rules. Information about the policy effects are logged, but not enforced.

Note: WSRM only applies processor resource allocations when there is contention for the resource. This means, although a process might have a 25% processor allocation, that process might use 100% of the processor as long as there are no other processes seeking processor time.

1.3 WSRM features

Figure 1-1 shows the main window of WSRM. It runs as a Microsoft Management Console plug-in.


Windows System Resource Manager
Action
View
Tree
Windows System Resource Manager (X445)
Resource Allocation Policies
Process Matching Criteria
Calendar [Enabled]
Resource Monitor
Accounting [Enabled]

Microsoft®
Windows® System Resource Manager

Current WSRM Status

WSRM management is currently RUNNING .
Click the status icon to change its settings.

Calendar - Enabled

Management Type - Manage

Current Resource Allocation Policy - Weekday

Notification - Disabled

Accounting - Enabled

Windows System Resource Manager Help

For context-sensitive help through out WSRM, press F1.

MANAGEMENT RUNNING

Figure 1-1 WSRM main window

1.3.1 Process matching criteria

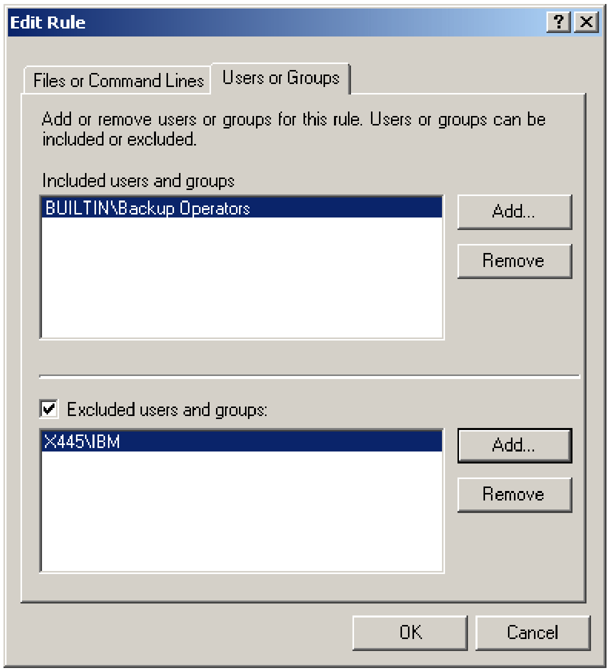
Services, processes, or applications that are to be managed by WSRM must be defined using process matching criteria. This criteria specifies the logic used to match and group processes. When a process is started, WSRM tries to match that process with process matching criteria. If it finds a match to the process, the process will then be managed according to the managing resource allocation policy with which it is associated, and rules specified in the resource allocation policy are applied.

If a process starts and is not matched with any of the process matching criteria, the process is automatically managed by the default WSRM group, unless the process is in the system exclusion list (which consists of operating systems services, and so forth, that are excluded from management) or the user exclusion list, and the policy's management mode has been set as “active”. The default group is allocated any CPU bandwidth that is not in use by the managing policy. This bandwidth is provided on an equal-share basis. By using a default group, WSRM can ensure that processes not defined by the managing policy will get equal access to the resources available to the default group.

WSRM provides the ability to use Windows group and user accounts when matching processes and allocating resources. During the setup of new process matching criteria, the Users or Groups tab enables the association of a process to a user, or to a member of a Windows group. When doing this, resource allocations are assigned and managed in the context of the user or member of the group that started the process. When using the Windows groups to associate a process, any member of that group will have system resources governed and allocated by WSRM (to the process that was started) because of the group membership they hold.

WSRM also includes the ability to add exclusions for members of groups. For example, you can create a process matching criteria and then specify that when a member of the Backup Operators group starts a process, they will be governed by rules that were applied during the process matching criteria because they are a member of this group. If another user starts the same process (and they are not a member of the Backup Operators group), they will be exempt from any of the rules specified in the resource allocation policy to that process matching criteria. The ability to add further exclusions to these groups is useful if a member of the Backup Operators group requires an exemption to the enforced rule by the managing resource allocation policy.

As shown in Figure 1-2, included and excluded users or groups can easily be included into a rule.



Add or remove users or groups for this rule. Users or groups can be included or excluded.

Included users and groups

BUILTIN\Backup Operators

☑ Excluded users and groups:

X445\IBM

Figure 1-2 Including and excluding users and groups to process matching criteria

1.3.2 Exclusion lists

There are some processes that should not be managed by WSRM. Taking resources away from these processes can cause problems and undesirable effects to the function of the operating system.

WSRM provides two lists where processes are excluded from resource management and accounting:

- System-defined exclusion list — A list of operating system-related processes. This list cannot be modified and is created by default when WSRM is installed.
- User-defined exclusion list — A list of processes, initially empty, that when defined, lets the administrator exclude other processes from management by WSRM.

Processes in either of these exclusion lists have the ability to consume resources without restrictions imposed by WSRM.

1.3.3 Resource allocation policies

Using resource allocation policies enables WSRM to specify the resource usage for a process. The first step to create a resource allocation policy is to create process matching criteria. The process matching criteria is then added to a resource allocation policy that then defines one or more resource allocations.

> **Tip:** WSRM ships with a default policy that allocates equal resources to all processes. Many customers might find this default policy sufficient.

To create a new process matching criteria from within the resource allocation policy window, right-click **Resource Allocation Policies** and click **New Resource Allocation Policy**. From the window that opens, click **Add** → **Genera**l → **Process Matching Criteria** → **New** → **Add**.

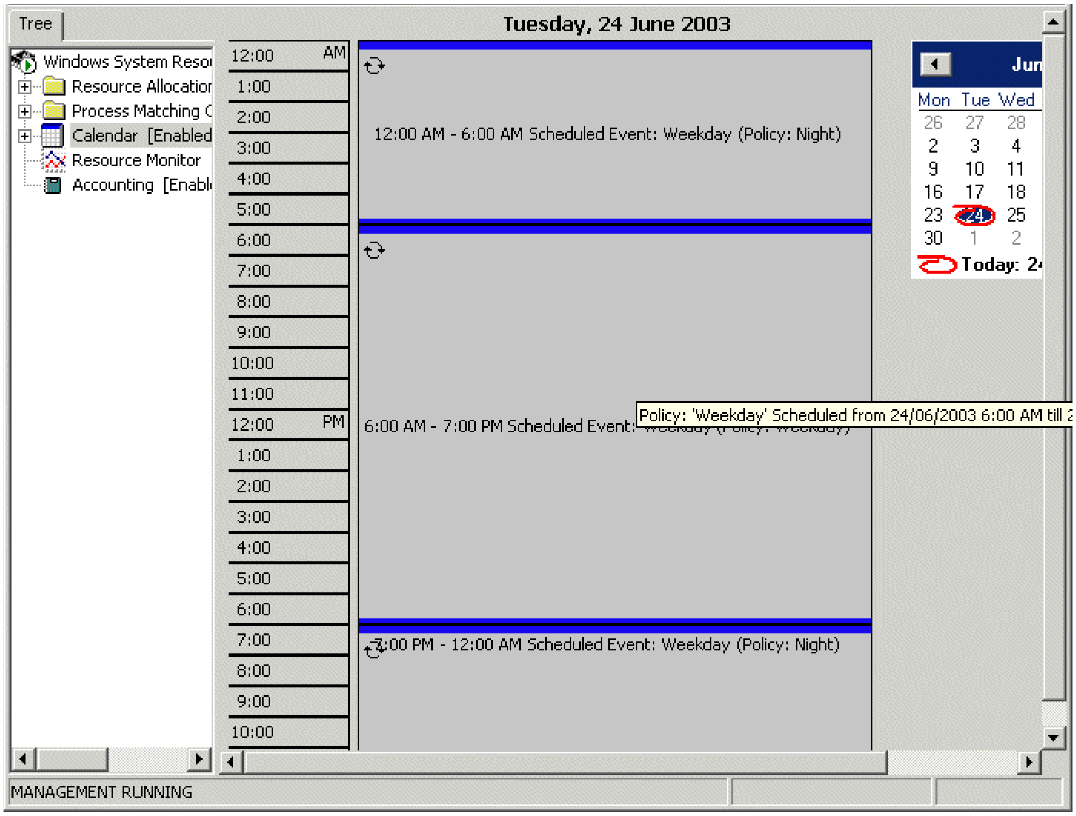
WSRM provides the following capabilities for managing the CPU and memory resource allocations for applications, services, and processes:

- Processor resource management, which can be achieved in the following ways:
  - Allocating processor bandwidth to an application, service, or process. When doing this, you set processor bandwidth limitations. Utilization is not allowed to raise above the set limit in the policy.
  - Processor affinity. When you set, using a resource allocation policy, processor affinity between a processor (or range of processors) and an application, service, or process they will always use the processor or processors to which they have been affinitized.

  > **Tip:** WSRM manages the full CPU and memory resources of the server, except for those resources reserved by the operating system for special system functions. As a result, the 100% that WSRM “sees” is, in fact, less than the full set of server resources due to these reserved resources. In other words, an allocation of 100% CPU in a resource allocation policy corresponds to the CPU capacity available on the system after the processes in the exclusion list have taken all the CPU they request.

- Managing memory resources, which can be achieved in the following ways:
  - Managing a process’s *working set*. A working set is the maximum number of pages (of the total virtual memory of a process) that can occupy physical memory at any point in time. WSRM sets the OS Working Set Size parameter as specified in the resource allocation policy. When the working set limit specified has been reached, the operating system simply swaps out existing pages from physical memory to the disk pagefile.
  - Managing a process’s committed memory size. Committed memory either resides in the working set of a process or maps to a pagefile. Committed memory is the total amount of memory allocated to a process. The actions possible when the commit charge is exceeded is to either log an event or terminate the process.
- WSRM calendar and scheduling

  The WSRM calendar provides the ability to schedule the automatic activation of a resource allocation policy. Resource allocation policies can be entered in the calendar to run once, or on a regular basis by defining a recurring calendar event, as shown in Figure 1-3.



12:00 AM - 6:00 AM Scheduled Event: Weekday (Policy: Night)

6:00 AM - 7:00 PM Scheduled Event: Weekday (Policy: Weekday)

Policy: 'Weekday' Scheduled from 24/06/2003 6:00 AM till 2

7:00 PM - 12:00 AM Scheduled Event: Weekday (Policy: Night)

Figure 1-3 WSRM Calendar view

The WSRM scheduler allows multiple resource allocation policies to run at different times over a 24 hour period by entering them into allocated time slots and then defining the schedule in the calendar.

Tip: In the calendar, you are able to associate either a policy name or a schedule name to run on the date and time you entered:

- When using a policy name to schedule the calendar, the policy that was specified will run between the dates and times entered in the calendar.

When specifying a schedule name, any policy that has been associated to that schedule can run at different, predefined times within any 24 hour period using the dates specified within that calendar event.

1.3.4 Resource Monitor

WSRM provides performance objects for System Monitor. This provides the ability to measure the performance of the WSRM system and other systems on the network. This real-time view of system performance can be saved and viewed at a later date and can also be saved as an HTML file.

Tip: Because this is the standard System Monitor interface, you can also incorporate any other performance objects and counters in your performance analysis.

WSRM Resource Monitor includes three specific WSRM performance objects and counters:

- WSRM policy

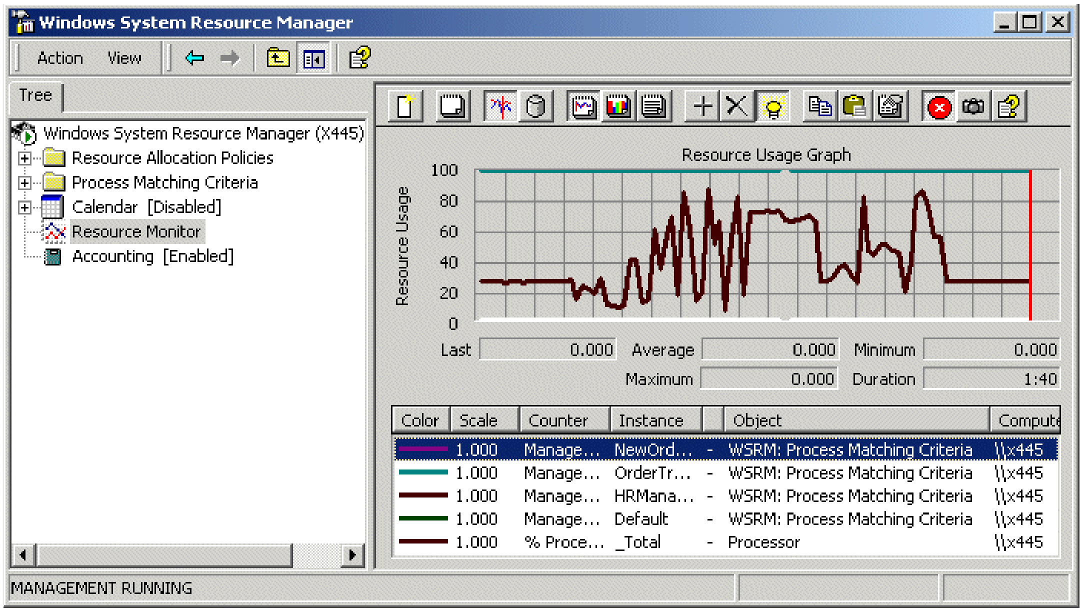
  The WSRM policy represents the percentage of CPU used by all processes managed with the current policy.

- WSRM process

  The WSRM process counter is a percentage of the total managed CPU used by a particular process since the resource allocation policy was set to manage the system.

- WSRM process matching criteria

  The WSRM process matching criteria provides granular performance measurement using several counters for each process matching criteria within the current managed resource allocation policy.



Windows System Resource Manager (X445)
- Resource Allocation Policies
- Process Matching Criteria
- Calendar [Disabled]
- Resource Monitor
- Accounting [Enabled]

Last 0.000 Average 0.000 Minimum 0.000
Maximum 0.000 Duration 1:40

Color	Scale	Counter	Instance		Object	Compute
🟪	1.000	Manage...	NewOrd...	-	WSRM: Process Matching Criteria	\\x445
🟦	1.000	Manage...	OrderTr...	-	WSRM: Process Matching Criteria	\\x445
🟥	1.000	Manage...	HRMana...	-	WSRM: Process Matching Criteria	\\x445
🟩	1.000	Manage...	Default	-	WSRM: Process Matching Criteria	\\x445
🟥	1.000	% Proce...	_Total	-	Processor	\\x445

MANAGEMENT RUNNING

Figure 1-4 WSRM Resource Monitor window

1.3.5 Accounting

Accounting is a feature of WSRM that logs information based on the behavior of managed processes. The information logged is stored in the accounting database:

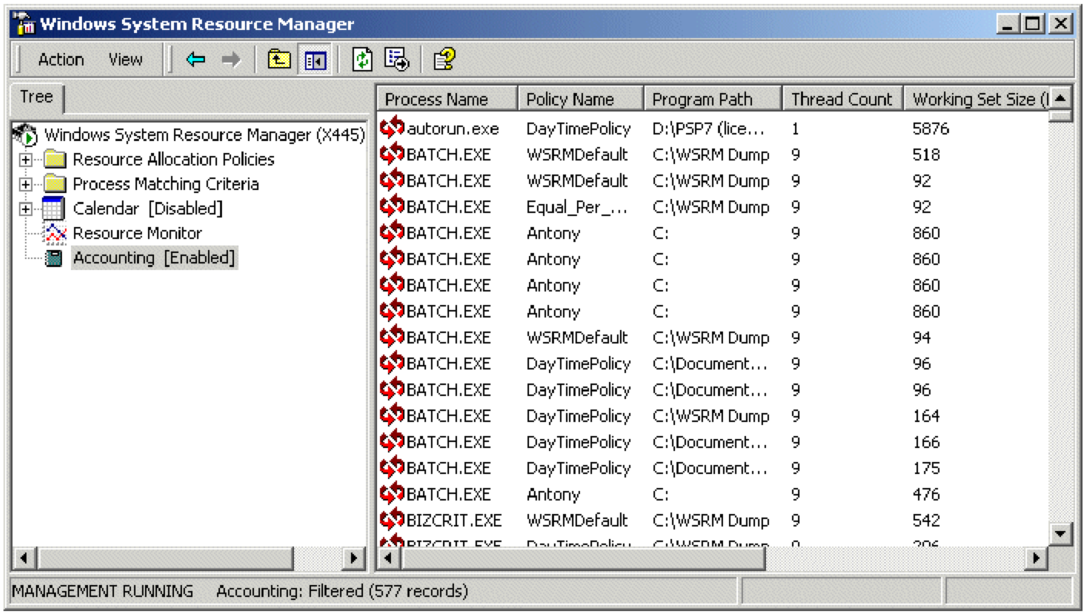
```
%windir%\system32\Windows System Resource Manager\JetDb\wsrm.edb
```

By default, information is logged into the database every 10 minutes, and this setting can be modified.

The information can be saved to a text file or as a CSV file. WSRM can also create an SMF formatted record for accounting and performance planning. SMF is the format used for resource accounting by applications such as those from Merrill Consultants (MXG), SAS, or Computer Associates (MICS). This information can then be used to generate reports on resource usage or to facilitate departmental resource charge back accounting. In other words, with this tool, a department could be charged depending on the memory and CPU resources they used.

The accounting database can also be copied through Windows Volume Shadow Copy. That won't resolve the need to stop WSRM at some point to reset the database space to empty, but it would allow data to be exported to accounting and performance tracking systems.

The information logged in the database pertains to processes that are defined within the process matching criteria of a resource allocation policy that is set to either a managing state or a profiling state. Accounting data always pertains to a specific policy. See Figure 1-5.


Windows System Resource Manager
Action
View
Tree
Windows System Resource Manager (X445)
Resource Allocation Policies
Process Matching Criteria
Calendar [Disabled]
Resource Monitor
Accounting [Enabled]

Process Name	Policy Name	Program Path	Thread Count	Working Set Size (I
autorun.exe	DayTimePolicy	D:\PSP7 (lice...	1	5876
BATCH.EXE	WSRMDefault	C:\WSRM Dump	9	518
BATCH.EXE	WSRMDefault	C:\WSRM Dump	9	92
BATCH.EXE	Equal_Per_...	C:\WSRM Dump	9	92
BATCH.EXE	Antony	C:	9	860
BATCH.EXE	Antony	C:	9	860
BATCH.EXE	Antony	C:	9	860
BATCH.EXE	Antony	C:	9	860
BATCH.EXE	WSRMDefault	C:\WSRM Dump	9	94
BATCH.EXE	DayTimePolicy	C:\Document...	9	96
BATCH.EXE	DayTimePolicy	C:\Document...	9	96
BATCH.EXE	DayTimePolicy	C:\WSRM Dump	9	164
BATCH.EXE	DayTimePolicy	C:\Document...	9	166
BATCH.EXE	DayTimePolicy	C:\Document...	9	175
BATCH.EXE	Antony	C:	9	476
BIZCRIT.EXE	WSRMDefault	C:\WSRM Dump	9	542

MANAGEMENT RUNNING Accounting: Filtered (577 records)

Figure 1-5 WSRM Accounting window

Unlike the Windows event log, the accounting database consumes as much disk space as is necessary; this, of course, is dependent on the disk storage space available. WSRM is not able to overwrite older records or purge the database as space runs out.

It is important to archive and delete the records in the accounting database on a regular basis so that the size of the database is continually managed. Microsoft recommends that the accounting database be located on a disk other than the one where the operating system is installed. This ensures that if the accounting database consumes all the disk space, the operating system continues to function.

All WSRM operations can be scripted and scheduled. This is particularly relevant to the maintenance of the accounting database by archival or deletion of old records, or both.

The following command is available to change the various settings in relation to the accounting database. The `/dbpath:path` command line option relocates the database file to the path specified.

```
wsrmc /set:config /acc [/dbpath:path] [/loginterval:value] [/timeout:value]
[\\server_name]
```

More information about the delete command line options can be found using WSRM help.

2

Positioning

This chapter explains the purpose and capabilities of Windows System Resource Manager (WSRM) and how it compares to other resource managers.

The need for resource management and the capabilities it provides when trying to achieve this are discussed in the previous chapter. Section 2.1, “The capabilities of WSRM” on page 12 describes the abilities of WSRM and its limitations.

Because WSRM is not the only resource management solution available in the marketplace, we briefly discuss other popular tools that achieve a similar result in 2.2, “Other resource management tools” on page 13.

Server consolidation is an important concept for customers when trying to reduce costs. WSRM plays a significant role in server consolidation. Therefore, we also discuss its role in this in 2.3, “Server consolidation” on page 14.

2.1 The capabilities of WSRM

WSRM has the ability to manage both CPU and memory resource allocations for applications, services, and processes running on a server. WSRM is supported on Windows 2003, Enterprise Edition and Datacenter Edition. It provides the ability to dynamically manage system resources. For systems that might otherwise struggle to complete tasks due to an inefficient use and allocation of CPU and memory resources, WSRM provides all the necessary tools in an easy and intuitive interface.

As with all performance-related issues in server environments, proper, consistent, and complete monitoring of all resource components is essential to identify bottlenecks before they occur. It is widely stated that Microsoft 32-bit server operating systems typically run short of memory before other system resources. WSRM can automatically and specifically manage the allocation of CPU and memory resource for most applications, services, and processes that are not otherwise included in system or user exclusion lists. These lists are introduced in 1.3, “WSRM features” on page 3.

WSRM is not, however, intended to handle the following data center functions:

- WSRM cannot manage resources across multiple servers.
- WSRM can run on cluster nodes to manage applications, but resource allocation policies active on a failed cluster node will not be automatically copied across to the failover node. Manual intervention is required to achieve this. In other words, WSRM is not currently cluster aware.
- WSRM cannot start or schedule applications or services.

Operations management software (for example, IBM Tivoli®, Microsoft Operations Manager) can be used to overcome these limitations. For example, operations management software can enable the deployment of WSRM resource allocation policies across a server estate, or schedule the execution of processes.

WSRM does not have the ability to affinitize, monitor, or manage the following components when resource contention occurs:

- Network interface

  Most IBM @server xSeries servers have at least one, normally two, integrated gigabit network interfaces cards (NIC). These NICs provide the server with redundant network connectivity. Additional NICs can be installed when more bandwidth is required. Server components such as processor, disk, and memory all have an impact on the performance of the NICs. Applications, drivers, and network designs can also affect overall server utilization of the network bandwidth.

- Disk subsystem

  All data must be read from, and sent to, the disk subsystem. Because disk operations are far slower than that of PCI transfers, memory accesses, and LAN transfers, the disk subsystem is a common source of bottlenecks in servers. An insufficient amount of memory in a server can cause excessive use of the pagefile that is stored on disk, thus causing a common disk I/O bottleneck. WSRM can manage the use of system memory, but excessive paging can still occur if the correct amount of memory is not installed. The easiest way to reduce paging is to install more physical memory.

- PCI devices

  All data that enters and leaves the server travels through the PCI bus. The performance of the PCI bus is defined by several factors, including the PCI adapters, the device drivers, and the system PCI chipset. The PCI bus will run at a speed of the slowest adapter installed in the slot on that PCI bus. Careful consideration should be taken when planning the installation of PCI adapters into a system, including the speed of the slot that the adapter is installed into and also comparing it to any other adapters installed on the same PCI bus. The xSeries 445 server has multiple PCI-X buses, so this is less of a problem.

Note: While PCI is discussed above in concept, PCI-X is a new PCI bus specification that is available throughout the range of xSeries servers. PCI-X provides greater data throughput and reliability for PCI-X compatible adapters. PCI-X adapters are also backward compatible with conventional PCI-designed systems. However, when placing a conventional PCI device onto a PCI-X capable bus, all devices on that bus will operate at conventional PCI speeds. Ethernet adapters do not benefit from PCI-X because they are limited by the buffer copies required by TCP/IP.

2.2 Other resource management tools

There are various tools available in the marketplace to provide similar functions to WSRM. This section briefly discusses and compares IBM Process Control and Aurema Application Workload Manager (AWM) with Microsoft Windows System Resource Manager.

2.2.1 IBM Process Control

Process Control is a process management tool for Intel®-based systems. It is no longer available for download from IBM. It helped improve performance both for the server and for application users in general. On a server, one application might have its priority set to a unnecessarily high level, becoming a detriment to all other applications. These types of ill-behaved applications made it virtually impossible to run many applications concurrently on a server.

Process Control was able to organize and manage processes and system memory resources on servers running Windows 2000. Process Control was developed by IBM and built into the Windows 2000 Datacenter Server offering.

For IBM @server xSeries customers, however, Process Control was also available for use on:

- Windows 2000 Server
- Windows 2000 Advanced Server

For Windows 2000, Process Control provided the following capabilities:

- Manage two or more processes as a group
- Server consolidation by the use of CPU affinity and memory constraints
- Secure servers from unauthorized applications or processes
- Resource utilization reporting and billing support

Process Control was designed to complement Windows 2000 Task Manager and System Monitor, but was not designed to replace them. To uniquely identify processes (using process control), the processes were assigned aliases. This is achieved by creating process alias rules. Once assigned aliases, processes can be arranged into groups, and rules can then be applied equally to all members of the group.

Process Control is no longer available. Development of this resource management tool has been discontinued.

2.2.2 Aurema Application Workload Manager

Aurema is a company that develops its own resource management software products, including Application Workload Manager (AWM), a plug-in to IBM Director 4.1.

While WSRM provides a comprehensive solution for resource management, it is available only for servers running Windows 2003 Enterprise and Windows 2003 Datacenter Editions. If additional support is required for Windows 2000 operating systems or Windows Server 2003, Web Edition and Standard Edition, then AWM from Aurema might be more suitable. It supports all Windows 2000 and all Windows Server 2003 family members. There are plans to support Linux also.

Like WSRM, AWM can manage and control resources, including CPU and physical system memory. When virtual memory is being overused, WSRM can either end the process consuming too much of the virtual memory, or it can log an entry to the event log while allowing the process to continually consume it. This might be suitable for most environments, but AWM has the added ability to restrict virtual memory usage of processes to fixed levels without terminating them.

Because AWM is a plug-in to IBM Director, customers who already use this systems management tool might find AWM more convenient to use.

For more information about AWM, see:

`http://www.ibm.com/servers/eserver/xseries/systems_management/director_4/awm.html`

2.3 Server consolidation

Recently, most companies have seen a proliferation of Intel-based servers. This has been caused primarily by the following:

- Distributed computing models

  In the 1970s, we saw the use of centralized computing in the form of the mainframe. As the mainframe matured, it introduced the use of sophisticated tools for remote mainframe client connectivity, centralized management, and centralized control. For the 20 years or so following this, business trends changed to a more distributed model. Information Technology followed these business trends by assigning IT groups to match the distributed business model. By doing so, this brought with it a proliferation of IT hardware and equipment that suited the individual business needs of the distributed business department.

- Inexpensive server hardware

  With the introduction of Intel-based hardware, these distributed business models were well matched with the low cost and relative ease of management the Intel-based servers offered. With an increased choice of software and applications that were available from independent software vendors (ISVs), the need to purchase a separate server to suit an individual business requirement was inexpensively met by using an Intel platform. Applications and software generally required specific, dedicated server hardware to run effectively and reliably, forcing businesses to new, additional servers, thus bringing with it additional costs and administrative overhead.

This proliferation of Intel-based hardware (including the advances in server and infrastructure technology) has lead to a shift in the relationship of the business and IT models. Most companies are still moving toward a more distributed business model, but they are realizing the cost savings and efficiency that can be achieved by moving their IT infrastructure models toward a more centralized approach.

Due to these reasons, companies are starting to realize the logical benefits that consolidation can provide. By consolidating IT infrastructures, the ability to lower the *total cost of ownership* (TCO) can be achieved. Server consolidation helps to achieve the following:

- Use server resources more efficiently. Rather than a server running at 25% capacity make it do more. Server consolidation provides the ability to raise the utilization of these servers to use some of the 75% remaining capacity.
- Raise server availability levels. Using large quantities of servers to accomplish lots of different functions introduces many points of failure. In short, the more servers you have, the greater chance of having potential system outages.
- Lower cost of administration. When server consolidation is deployed, an overall drop in the administration costs are also realized. By using a centralized model to provide IT services, the requirement for large numbers of IT human resources are also reduced.
- Lower the requirement of real estate. Also called site consolidation, by relocating servers to a fewer number of sites, this centralizes the computing and lowers the real estate costs to a business.

Server consolidation consists of the following subcomponents

- Data consolidation — Moving data to a single data repository to provide data security, disaster recovery, and better storage management.
- Physical consolidation — Reducing or replacing large quantities of servers with fewer, more powerful ones. Typically, they serve the same application or function, that is, file and print, Microsoft Exchange server, and so forth.
- Site consolidation — Centralizing and reducing the number of physical distributed locations to a fewer number of sites.
- Application consolidation — Hosting multiple applications on the same server hardware, also referred to as *application stacking*.

Of the above server consolidation subcomponents, WSRM plays a significant role in application consolidation. Therefore, we discuss this further in the following section.

2.3.1 WSRM and its role in application consolidation

Application stacking has not been a reliable approach for achieving server consolidation in the past. This is due to the inability of many applications to coexist within the same operating system environment. Applications running processes that consume as much available processor time as they are able, or applications that cause problems such as memory leaks, will typically starve server resources, thus causing the other applications to fail or to respond negatively.

As discussed in previous sections, applications start processes that are given a priority value. This value, in part, determines the order in which its threads, in relation to all other threads, are executed by the processor. But not all applications will run reliably together. ISVs generally design their applications to operate on a dedicated server, with dedicate processors and memory. It is impossible for ISVs to design their applications so that they will always coexist happily with any other application.

Microsoft developed WSRM to enable customers to achieve reliable and flexible application consolidation through the use of process-based workload consolidation. WSRM changes the priority given to each process, thus influencing the order in which these processes (and their threads) are executed by a processor. Also, having the ability to regulate and enforce working set and committed memory limitations, WSRM offers a robust, comprehensive, and reliable solution for resource management.

3

Planning

This chapter describes how to plan for WSRM implementation. This includes WSRM best practices and considerations when planning to implement resource systems management. We also discuss the implications of using the WSRM tool in a clustered or NUMA-based environment. This chapter provides the ideas and content for discussion prior to WSRM rollout.

This chapter includes the following topics:

- WSRM best practices
- Planning steps for WSRM
- Running WSRM in a clustered environment
- WSRM and NUMA support

3.1 Planning steps for WSRM

When implementing WSRM, it is important to discuss and plan its implementation. The rules and policies assigned in WSRM can potentially cause problems if careful resource allocation is not planned for appropriately.

The WSRM server component is currently supported on Windows Server 2003, Datacenter Edition and Enterprise Edition, both the 32-bit IA32 and 64-bit Itanium architectures. The WSRM client is supported on 32-bit versions of Windows 2000 SP3, Windows XP, and Windows Server 2003.

When planning for WSRM, consider the following:

- WSRM should be the only resource management tool managing the resources of the server.
- WSRM is unable to manage memory resources such as Microsoft Address Windowing Extensions (AWE) memory, large page memory, or locked memory. It cannot manage the allocation limits of paged and non-paged pool memory.
- If running multiple applications on a server, and you plan to use WSRM, it is especially important that you understand the application resource requirements and the functionality of WSRM.
- As discussed in 1.2, “How WSRM works” on page 2, WSRM profiling policies provide a way to view the implications of a policy without putting the policies rules into action. This can be a useful tool to aid your planning process.

3.1.1 An introduction to planning

The decision to use WSRM might be made early in the planning stages of a project, or it could be after a server has been deployed and placed into a production environment.

The full potential of WSRM can be demonstrated after it is deployed into a production environment. WSRM assists servers in reaching greater performance levels due to its dynamic resource allocation and resource re-allocation abilities for most applications, services, and processes.

If you are in the early stages of a project, you are better positioned for success. This allows for planning and installation of a server by deciding on the best and appropriate hardware and the applications that will run on the server.

3.1.2 Planning server and application consolidation

Some common project planning questions are:

- What are the requirements and the expectations for this requirement?
- Are performance requirements understood or documented?
- What should the server hardware configuration be?
- What level of service and support will be required?
- What software configuration will be needed? Applications, and so forth?
- What are the installation plans?
- Do I have plans to implement change control? What are those plans?
- Do the technical staff have the technical skills required to implement this project?

- Are there additional education or consulting services required?
- Do I have a suitable environment to implement this?
- What disaster recovery plans are going to be used?
- What are the migration plans for the hardware, software, and data?
- Does the proposed solution meet requirement criteria?

3.1.3 Performance and server components

After a requirement for a new application has been identified, the appropriate server hardware to host that application must also be found. Most applications have a minimum hardware requirement, such as the speed of the processor or the amount of physical memory required to ensure that the application performs properly. The recommended hardware requirements for the software should not be overlooked. It is also be necessary to understand the behavior of the application, and therefore, it might be necessary, if possible, to view the application under stress in a pilot environment before migrating it into production.

Generally speaking, if an application does not behave properly or reliably on a server, overall server performance is a very difficult thing to achieve. For example, if an old application consumes too much system memory or continually exhibits memory leaks, performance of the server and the other applications will suffer. Resource management will be near impossible to control without the use of WSRM in these environments.

Hardware planning

Planning for the appropriate hardware includes the following points:

- Processors
  - Choose the most appropriate processors for the application.
  - Ensure that the application supports the quantity of processors installed within the server hardware.
  - Would larger processor cache improve the applications performance, that is, database applications?
- Should Hyper-Threading be turned on or off?

  Hyper-Threading is technology in Xeon MP, Xeon DP, and Pentium® 4 processors that allows a single physical processor to execute two separate code streams, or threads, concurrently. To the operating system, a physical processor appears as two logical processors. Hyper-Threading is designed to exploit the multithreaded capability of the applications and the operating system.

  Fewer processors generally provide the best gains from Hyper-Threading technology. Tests in the xSeries performance lab indicate the following performance gains are likely:

  - Two physical processors: 15-25% performance gain
  - Four physical processors: 1-13% performance gain
  - Eight physical processors: 0-5% performance gain.

  With Hyper-Threading on, the WSRM calculation of available CPU might be different from the customer's expectation. Hyper-Threading is transparent to WSRM.

  For more information about Hyper-Threading, see:

  http://www.intel.com/technology/hyperthread/
- Memory
  - How much memory will the application and the operating system support?

  - Check and verify the application's minimal memory requirements as set by the ISV.
  - Disk paging is a common disk bottleneck and cause of performance degradation. The easiest way to solve and avoid this problem is to install enough memory to minimize the need to use the pagefile. The correct amount can be difficult to identify, so ensuring the server's hardware provides some room for growth is important.
- Disks
  - A SAN data storage is often advantageous when addressing better disk I/O performance, modular growth, ease of management, and centralized control.
  - Choosing the best disk configuration to start with is essential and is often very difficult to change after a server has been placed into production.
  - Including several other factors, the size and speed of the drives significantly affects disk I/O performance.
  - Ensure the best IBM FAStT product is selected to meet the business requirements today, and into the future.
- Networking
  - If the network backbone is 10 Mbps, using gigabit network adapters do not increase network performance. Applications that generate large network I/O will struggle to meet user requirements if the network is a bottleneck.
  - Using multiple network cards increases the network performance of the application. If using gigabit NICs, ensure that the network infrastructure will support these speeds.
- Growth
  - Ensure that the server chosen provides sufficient growth capacity and a good return on investment.

Application planning

Planning for the application requirements includes the following points:

- If business demands on the application increase, ensure that the application can scale to support the server hardware.
- What disk layout is optimal for the application?
- What processes will the data perform?
- What does the application vendor recommend to improve performance?

3.1.4 Planning example

As an example, let's say you have three programs that will run on a server:

- A program that starts at 8 p.m. every evening to batch process data that was collected all day.
- A DB2® database that is read and written to during the business hours of 7 a.m. to 6 p.m. Monday to Friday. Customer orders are written to this database. It is most important that this application be available and responsive above all others.
- An internal application that allows employees to order supplies. While it is important, it is a low priority compared to the customer order database.

You would want a policy that gives priority to the customer order database during normal business hours. You would want to give evening priority to the batch process that runs from 8 p.m. to 11 p.m. You would want to make sure the employee supply order program is not permitted to run excessive resources at any time. It must always be subordinate to both the daytime customer order program and the evening batch processing.

As part of the WSRM planning process, create a table to document your goals. List all policies, detailing applications that will run within each policy, document CPU allocations, time and day to run, and so forth. This will allow you to plan your resource management effectively. Add columns to meet your needs.

Table 3-1 Sample planning documentation

Policy	Process/app	Day/time	CPU%	Comment
Daytime	OrderEntry	M-F/6a.m.-6p.m.	50	
	HumanRes	M-F/6a.m.-6p.m.	20	
	Accounting	M-F/6a.m.-6p.m.	10	
Evening	BatchJob1	M-S/6p.m.-6a.m.	60	
	BatchJob2	M-S/6p.m.-6a.m.	10	
Weekend	Accounting	M-F/6a.m.-6p.m.	50	
	HumanRes	M-F/6a.m.-6p.m.	30	
	OrderEntry	M-F/6a.m.-6p.m.	10	

3.2 WSRM best practices

Consider the following guidance to maximize the benefits of WSRM:

- Windows System Resource Manager should not be used in conjunction with another resource manager.

  There are many types of resource managers (for example, Aurema Armtech), and some are built into programs whose primary function is not resource management. Do not use other resource managers on the same system. Like many tools, you need to pick the one best suited for your situation.

- CPU resource allocation should not exceed 95%.

  As you allocate CPU resources, you should leave some CPU resources available to unmanaged applications or applications that do not have a resource allocation policy. Note that the exclusion list processes (system or user) are always satisfied first before enforcing a policy.

- Use caution when setting processor affinity.

  Do not set processor affinity unless you understand the impact of the changes you make. It is just as easy to cause performance degradation as it is to boost performance.

- Regularly archive and delete accounting data.

  Accounting data is used for many purposes, including assessment of the system in terms of performance bandwidth, comparing performance, or possibly charge-back accounting. Once enabled, the size of the database continues to grow until space fills up or the database is archived. WSRM auditing does not overwrite old records. If you use the C: drive for auditing data, you could possibly run out of space needed by the operating system.

- Implication of adding processes to the user-defined exclusion list.

  If an application that hosts other applications (for example, the Microsoft Management Console, mmc.exe) is added to the user-defined exclusion list, it could potentially consume all available bandwidth because it will be an unmanaged process.

- Do not use Windows System Resource Manager to manage applications or processes that are self managing.

  If you must run these types of programs on the same server as the Windows System Resource Manager, add them to the user-defined exclusion list. Microsoft states that these applications will otherwise disrupt the algorithm that WSRM uses.

  For example, Microsoft SQL Server can manage its own processor affinity through the administrator interface so its resources should not be managed using WSRM. WSRM can be used to manage the SQL Server CPU consumption, but do not use it to control its memory or affinity settings.

- Processes running job objects cannot be managed.

  If an application (that is, Proccon or Sentinel) creates processes that use job objects, add these to the user-defined exclusion list. These objects should not be managed by WSRM. Refer to the documentation to see if an application uses job objects.

- Monitor all applications after limiting the resources of any single application.

  Applications can have complex or unclear relationships. If you limit memory usage on one application, it could adversely impact a different application with which it interacts. After making changes to resources, you need to monitor performance of all the applications through peak periods.

- Create a table that documents your goals.

  As part of the WSRM planning process, create a table to document your goals. List all policies, detailing applications that will run within each policy, document CPU allocations, time and day to run, and so forth. This enables you to plan your resource management effectively.

3.3 Running WSRM in a clustered environment

WSRM manages individual machines and is not cluster aware. As a result, cluster nodes must be managed separately. For ease of management, Microsoft recommends (but does not require) the same policy be used on each of the cluster nodes.

When Node A fails over to Node B, WSRM does not automatically fail over the managing policy from the failing node to Node B. If WSRM is to be used in a cluster, understand that there can often be several failover scenarios. You might have a cluster Node A and Node B that can both fail over to Node C. The failover or standby node could end up running different combinations of applications. Then again, you might have several identical servers that fail over to the same spare clustered node. In that circumstance, you might make all of your WSRM policies different to suit different cluster requirements.

To ensure consistent workload balancing regardless of the state of all nodes, we recommend that you create a policy that provides for some or all applications running in the cluster to run all on just one node. To ensure identical policies, WSRM lets you import and export either through the management console or from a command line interface.

To export or import from the management console, right-click **Windows System Resource Manager** in the tree pane. You have the option to import or export WSRM information in the form of XML files. This enables you to move the resource allocation policies and other potentially needed configuration data to a target server that might need to be active in a cluster failover. When importing into the target server, you will be prompted to click **Yes** to overwrite the existing WSRM data.

The command line interface offers the same functionality using the following syntax:

- To export WSRM information at the command prompt, use the command:

```
wsrmc /Export /Dir:<name>
```

- To import WSRM information at the command prompt, use the command:

```
wsrmc /Import /Dir:<name>
```

Note that in conjunction with operations management products, it is possible to create scripts that will deploy WSRM policies across a server estate depending on specific triggers. For example, should a failover occur, the operations management product could take care of installing new resource allocation policies across the remaining active nodes of a cluster to cater for new workload.

3.4 WSRM and NUMA support

Non-uniform memory access (NUMA) is an architecture designed to improve performance and solve latency problems inherent in large (greater than four processors) SMP systems. The IBM @server xSeries 445 server implements NUMA-based architecture and can scale up to 32 processors using multiple SMP Expansion Modules.

Windows Server 2003 includes new support and optimizations based on NUMA architectures. WSRM does not attempt to further optimize the applications’ use of NUMA systems.

4

Using WSRM

This chapter describes how to set up and use Windows System Resource Manager.

We discuss the following topics:

- Installing WSRM
- Starting WSRM
- Sample programs
- Setting up policies
- How WSRM handles resource contention
- Accounting
- CPU affinity and memory resource assignment

4.1 Installing WSRM

WSRM has two components:

- The WSRM server component runs on the server whose memory and CPU resources you want to control. This component is supported on Windows Server 2003, Datacenter Edition and Enterprise Edition, both the 32-bit IA32 and 64-bit Itanium architectures.
- The WSRM client is the user interface and can run either on the server or a supported remote workstation. Supported operating systems are 32-bit versions of Windows 2000 SP3, Windows XP, and Windows Server 2003.

WSRM is not installed by default. A separate WSRM CD-ROM is included with the operating system package. Insert the CD and follow the instructions to install the application.

Tip: If you are using an evaluation version of the operating system, download WSRM from:

`http://www.microsoft.com/windowsserver2003/downloads/wsrm.mspx`

When installing WSRM on Windows Server 2003, Datacenter Edition and Enterprise Edition, you are prompted about whether you want to install the server component or the client component, or both. The default is to install both. You are also prompted about whether you want to enable accounting. You can also enable or disable this later through the Accounting icon on the main window.

When installing WSRM on any other supported platform, you can only select the client component.

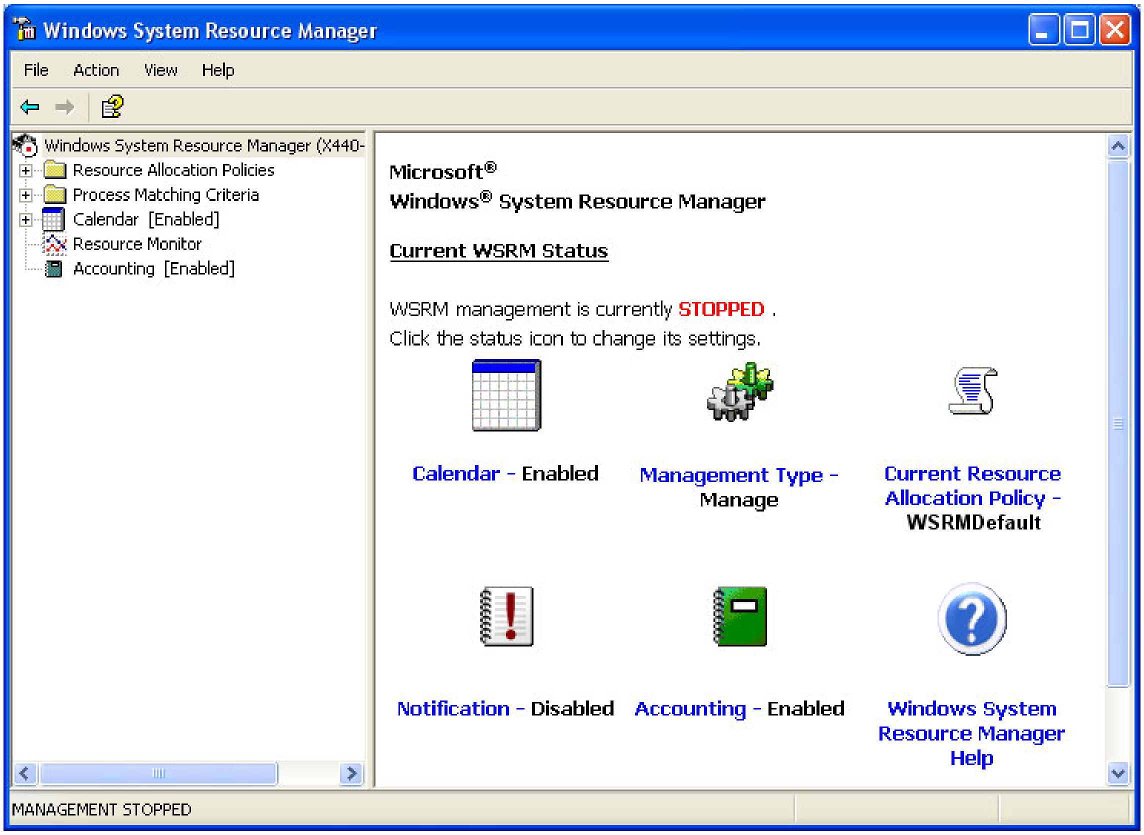
4.2 Starting WSRM

After WSRM is installed, start the user interface by clicking **Start** → **Programs** → **Administrative Tools** → **Windows System Resource Manager**.

The WSRM management console opens. The first time the WSRM client runs, it is not connected to the server component. To connect it to the server, right-click **Windows System Resource Manager (Disconnected)**, select **Connect to Computer**, and then enter the host name of the server you want to manage.

Note: You must be a member of the Administrators group on the server to connect.

After you are connected, the main window opens, as shown in Figure 4-1.



Windows System Resource Manager (X440-
- Resource Allocation Policies
- Process Matching Criteria
- Calendar [Enabled]
- Resource Monitor
- Accounting [Enabled]

Microsoft®
Windows® System Resource Manager

Current WSRM Status

WSRM management is currently STOPPED .
Click the status icon to change its settings.

Calendar - Enabled

Management Type - Manage

Current Resource Allocation Policy - WSRMDefault

Notification - Disabled

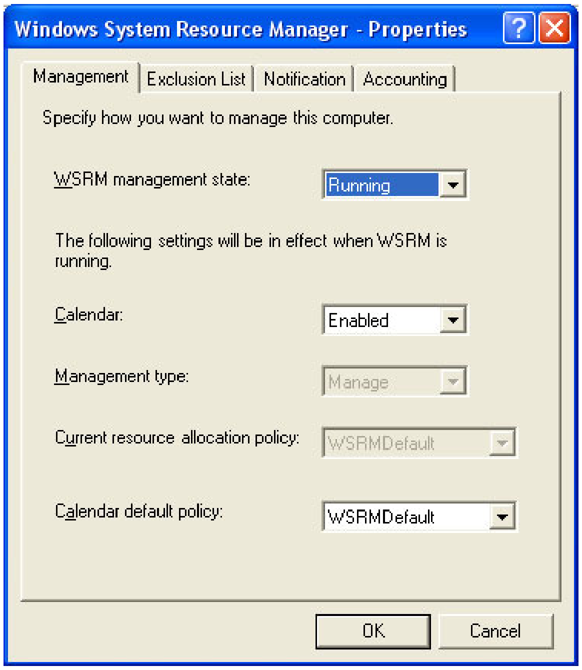
Accounting - Enabled

Windows System Resource Manager Help

MANAGEMENT STOPPED

Figure 4-1 Windows System Resource Manager main window

Initially, WSRM is stopped. To start it, click **STOPPED**. The properties window opens. Change the WSRM management state to **Running** and click **OK**.



Specify how you want to manage this computer.

WSRM management state: Running

The following settings will be in effect when WSRM is running.

Calendar: Enabled

Management type: Manage

Current resource allocation policy: WSRMDefault

Calendar default policy: WSRMDefault

Figure 4-2 Properties window where WSRM is set to Running

4.3 Sample programs

You can learn the basics of using WSRM with the sample programs that ship with the product. Microsoft included sample policies and programs that allow you to experiment with WSRM and see how different applications impact one another. In the samples directory (on the setup CD or directory), there are various sample files that help demonstrate the features and capabilities of WSRM.

You must import the sample XML files into WSRM before WSRM can use them. To import the sample XML files in Windows System Resource Manager, right-click **Windows System Resource Manager (servername)** → **Import WSRM Information** and select the location of the sample XML files.

The x86\samples directory contains the following four programs that display different characteristics and will each consume resources differently:

- Batch
- BizCrit
- Rogue
- Leaky

Read the samples.htm file for a full description of the XML files and executables.

4.4 Setting up policies

The easiest way to describe setting up policies is to show an example.

Tip: Online help is available in the Microsoft Management Console (MMC) by right-clicking any item in the left-hand tree panel and clicking **Help**.

The following example demonstrates WSRM resource management of multiple applications hosted on an IBM @server xSeries 445 server running Windows Server 2003, Datacenter Edition. Because there are multiple applications with various degrees of criticality, WSRM provides the necessary resource management.

Note: This example shows only the basics and concepts using WSRM. This Redpaper is not intended to explore the advanced resource management capabilities of WSRM.

4.4.1 Application requirements

The applications used in our example scenario are:

- NewOrderEntry (NOE.EXE)
- OrderTracking (OT.EXE)
- HRManagement (HRM.EXE)
- BatchBilling (BB.EXE)

Careful planning, as discussed in Chapter 3, “Planning” on page 17, is required to determine which applications need the most resources. This includes the CPU bandwidth that is required by each application, the memory usage of those applications, and at what days and times these resources are required.

Tip: Review Chapter 3, “Planning” on page 17 to provide the planning scope.

In this example, the following requirements have been determined for each application:

- NewOrderEntry

  The NewOrderEntry program needs to have the highest priority Monday through Friday, 6 a.m. to 7 p.m. Employees taking orders need the fastest response.

- OrderTracking

  Order entry personnel use OrderTracking, and this needs to be responsive. At peak periods, OrderEntry will be given more resources than OrderTracking. 24x7 customer service personnel can also use OrderTracking and, therefore, need high performance and availability of this program.

- HRManagement

  The Human Resources department invested 10% of the funds needed to maintain the server installation (an xSeries 445) and in return want to be guaranteed that they will receive 10% of the system resources during certain hours. They need the Human Resources Management application to be highly responsive from 6 a.m. through 7 p.m., seven days a week. Weekend work might also occur. This application is important, but customer- and revenue-related activities, including programs such as NewOrderEntry and OrderTracking, must have a higher priority.

- BatchBilling

  The BatchBilling application needs priority during the evening, seven days a week. Without this priority, BatchBilling would not complete in the allocated window of time.

Any programs not matched to the process matching criteria by WSRM, and not included in system- or user-defined exclusion lists will be managed equally by WSRM (with the remaining system resources).

4.4.2 Server policy worksheet

A worksheet, such as the one shown in Table 4-1, is useful to document and plan your resource allocation goals. This documentation is useful to help design the policies that WSRM will implement during the specific time periods. It is helpful to decide on the policy names, the applications to run, the times the managing policy will be in effect, the percentage of CPU to assign to each process, and finally, the working set and committed maximum memory levels that each process can consume.

Table 4-1 Server policy table

Policy name	Application	Description	Time	Days	CPU%
Weekday	NOE.EXE	OrderEntry	6a.m.-7p.m.	Mon-Fri	60
	OT.EXE	OrderTracking	6a.m.-7p.m.	Mon-Fri	25
	HRM.EXE	HRManagement	6a.m.-7p.m.	Mon-Fri	10
	Default group	All others	6a.m.-7p.m.	Mon-Fri	Remainder
Night	BB.EXE	BatchBilling	7p.m.-6a.m.	Mon-Sun	65
	OT.EXE	OrderTracking	7p.m.-6a.m.	Mon-Sun	15
	HRM.EXE	HRManagement	7p.m.-6a.m.	Mon-Sun	10
	Default group	All others	7p.m.-6a.m.	Mon-Sun	Remainder

Policy name	Application	Description	Time	Days	CPU%
Weekend	BB.EXE	BatchBilling	6a.m.-7p.m.	Sat-Sun	30
	OT.EXE	OrderTracking	6a.m.-7p.m.	Sat-Sun	25
	HRM.EXE	HRManagement	6a.m.-7p.m.	Sat-Sun	10
	Default group	All others	6a.m.-7p.m.	Sat-Sun	Remainder

Tip: WSRM provides 100% resource allocation among processes; however, up to 99% is available to be allocated by the administrator, not the total 100%. WSRM ensures that any processes, not matched using process matching criteria, are able to run with the remaining 1% of allowable CPU bandwidth.

If only 60% processor allocation is defined to all running processes, the unmatched processes will consume the resources that are available. Like any managing resource allocation policy, WSRM only enforces the minimum resource allocation to the default group when resource contention occurs.

4.4.3 Setting up process matching criteria

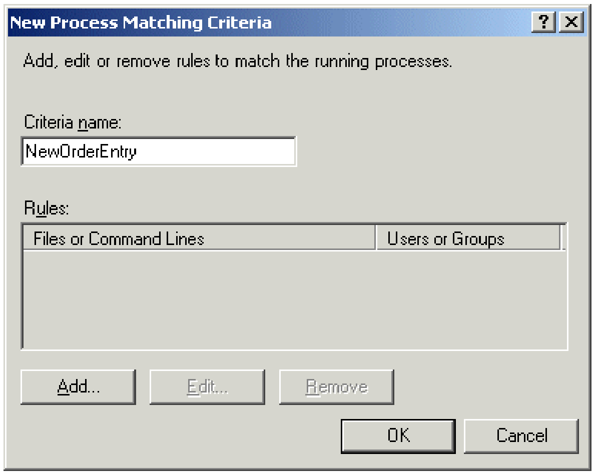
As discussed previously, the creation of process matching criteria and resource allocation policies is required to accomplish the goal of the policies in Table 4-1 on page 29. In the WSRM management console, process matching criteria are required for each application (also services or processes, if defined) that is entered in the server policy table.

The first step in customizing WSRM is to create process matching criteria for the applications defined in the server policy table. Applications typically have corresponding processes that are usually named similarly to the application itself. The applications (and processes) to be managed here are:

- NewOrderEntry (NOE.EXE)
- OrderTracking (OT.EXE)
- HRManagement (HRM.EXE)
- BatchBilling (BB.EXE)

To create the process matching criteria:

1. Within WSRM, right-click **Process Matching Criteria** and select **New Process Matching Criteria**.
2. For the name of the criteria, type `NewOrderEntry`. Spaces are not allowed in the criteria name field. Click **Add**. See Figure 4-3.



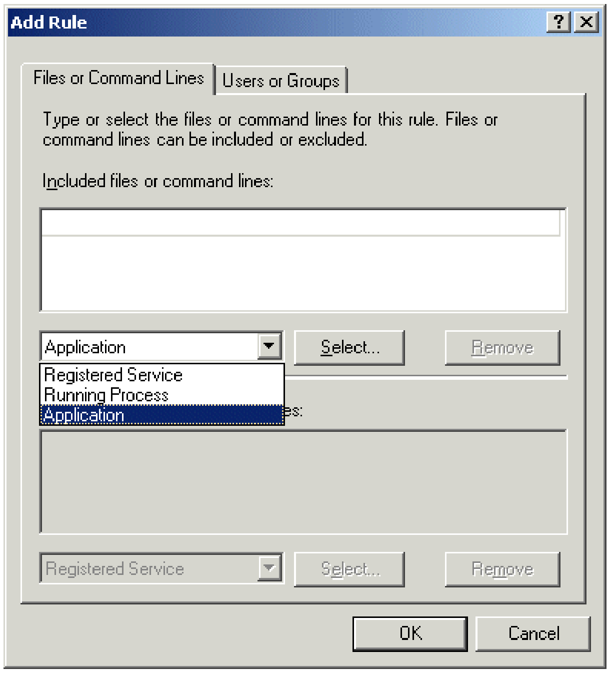
New Process Matching Criteria

Add, edit or remove rules to match the running processes.

Criteria name: NewOrderEntry

Figure 4-3 Entering the new criteria name

3. In the Add Rule window, select **Application** from the drop-down menu, as shown in Figure 4-4. Click **Select** and navigate to the program executable. In our example, this is NOE.EXE.



Add Rule

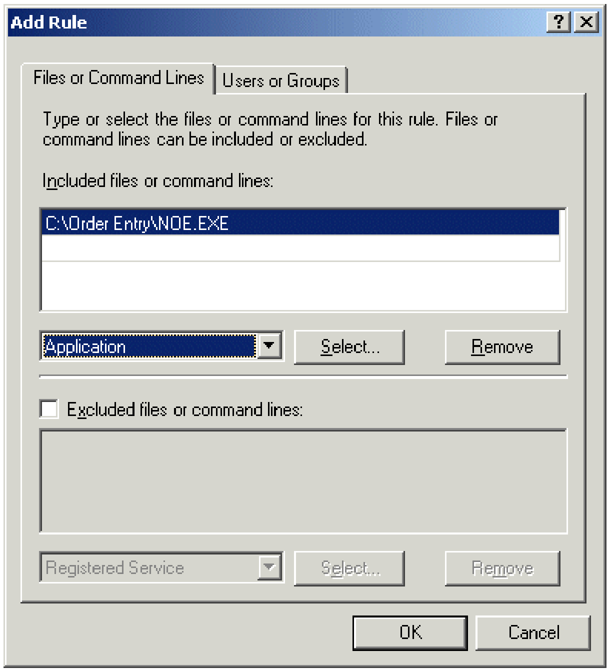
Type or select the files or command lines for this rule. Files or command lines can be included or excluded.

Included files or command lines:

Application

Figure 4-4 Selecting the type of file to run

4. Click **OK**. See Figure 4-5.



Add Rule

Type or select the files or command lines for this rule. Files or command lines can be included or excluded.

Included files or command lines:

C:\Order Entry\NOE.EXE

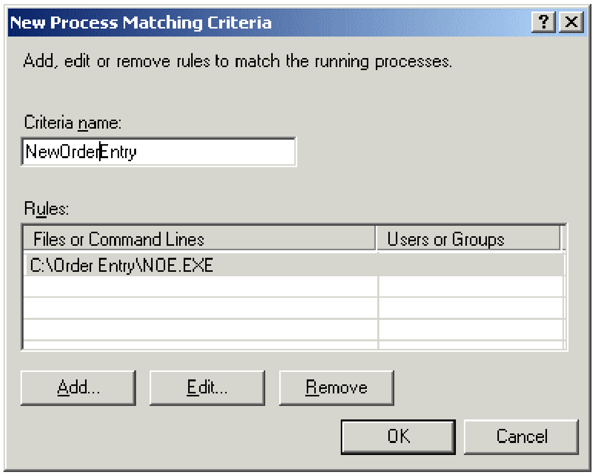
Application

□ Excluded files or command lines:

Registered Service

Figure 4-5 Enter the rule that will be run

5. Click **OK** to create the criteria. See Figure 4-6.



New Process Matching Criteria

Add, edit or remove rules to match the running processes.

Criteria name:

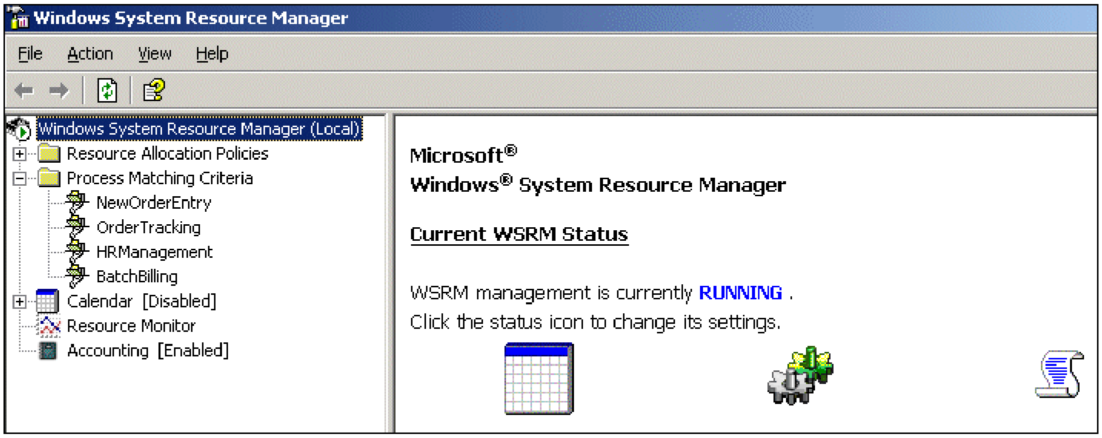
NewOrderEntry

Rules:

Files or Command Lines	Users or Groups
C:\Order Entry\NOE.EXE	

Figure 4-6 View of rules for one criteria

6. Create other process matching criteria for the other applications you want to manage. When complete, all of the applications will be listed in the left pane of the WSRM console under Process Matching Criteria, as shown in Figure 4-7.



Microsoft®
Windows® System Resource Manager

Current WSRM Status

WSRM management is currently RUNNING .
Click the status icon to change its settings.

Figure 4-7 Left pane shows new Process Matching Criteria

4.4.4 Setting up resource allocation policies

After creating the process matching criteria, it's now necessary to create a resource allocation policy that enables WSRM to match the processes to the managing resource allocation policy.

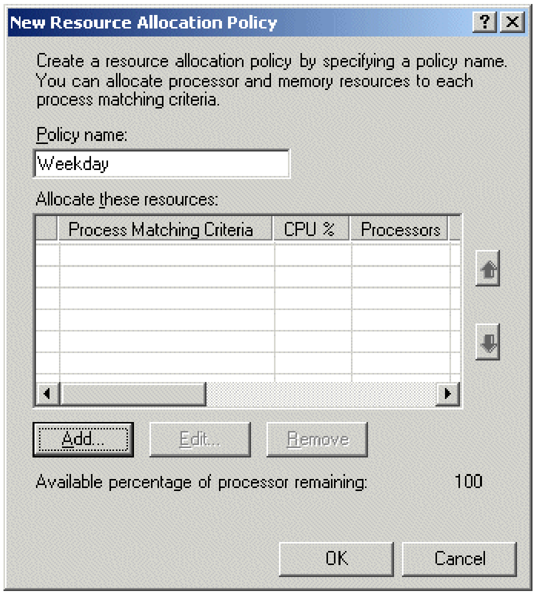
To create a new resource allocation policy, refer to the server policy table, Table 4-1 on page 29, and note the policy names specified. In our case, the policy names are:

- Weekday
- Night
- Weekend

To create the resource allocation policy, do the following:

1. Within WSRM, right-click **Resource Allocation Policies** and select **New Resource Allocation Policy**.
2. Enter the policy name. In our case, `Weekday`, as shown in Figure 4-8.

> **Tip:** Spaces are not allowed in the Policy name field.



Create a resource allocation policy by specifying a policy name. You can allocate processor and memory resources to each process matching criteria.

Policy name: Weekday

Allocate these resources:

Process Matching Criteria	CPU %	Processors

Available percentage of processor remaining: 100

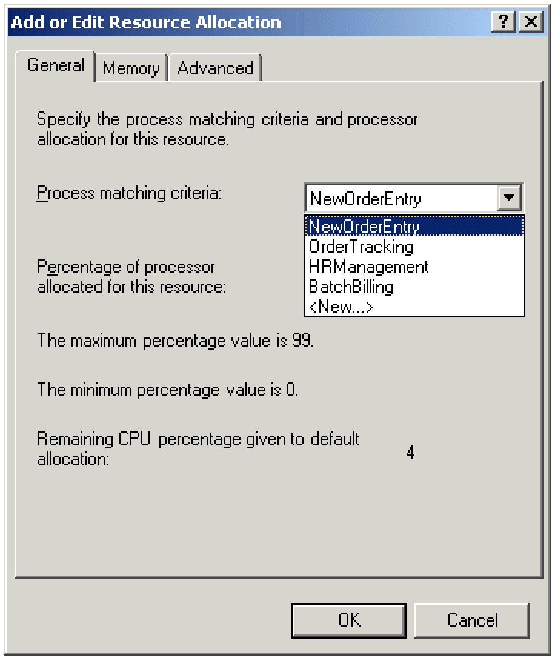
Figure 4-8 New Resource Allocation Policy window

3. Click **Add**.

4. Select each of the applications from the Process matching criteria drop-down menu, and enter the percentage of processor resources that are to be allocated through the Weekday policy. See Figure 4-9.

   In our example, the Weekday policy has three applications with the following processor bandwidth allocations:

   – NewOrderEntry (NOE.EXE) 60% CPU
   – OrderTracking (OT.EXE) 25% CPU
   – HRManagement (HRM.EXE) 10% CPU



Specify the process matching criteria and processor allocation for this resource.

Process matching criteria: NewOrderEntry

Percentage of processor allocated for this resource:

The maximum percentage value is 99.

The minimum percentage value is 0.

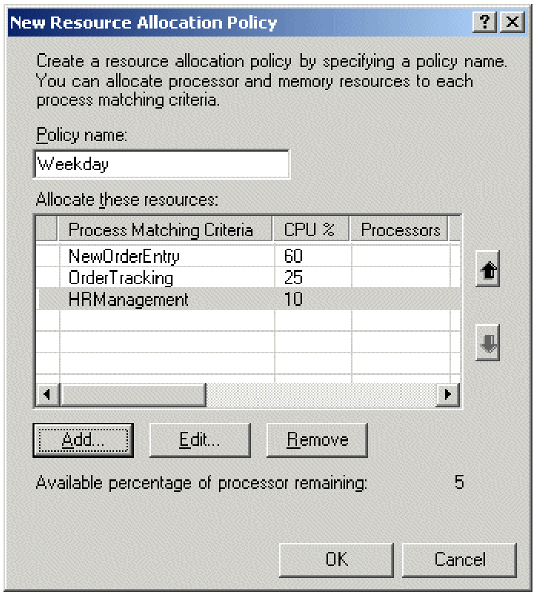
Remaining CPU percentage given to default allocation: 4

Figure 4-9 Adding a process matching criteria to the policy

5. Click **OK** and continue until all applications and their percentages are added. The Process matching criteria entries that have been created in 4.4.3, “Setting up process matching criteria” on page 30, appears in the drop-down menu.

> **Tip:** New process matching criteria can also be created from the drop-down menu by selecting **New**.

In our example, the Weekday policy looks like the policy shown in Figure 4-10.



Create a resource allocation policy by specifying a policy name. You can allocate processor and memory resources to each process matching criteria.

Policy name:
Weekday

Allocate these resources:

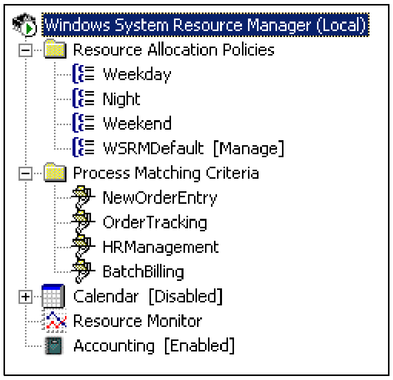
Process Matching Criteria	CPU %	Processors
NewOrderEntry	60	
OrderTracking	25	
HRManagement	10	

Available percentage of processor remaining: 5

Figure 4-10 Adding process matching criteria to resource allocation policy

6. Click **OK**.

7. Repeat this to create resource allocation policies for the remaining two processes. The WSRM management console should now display all three resource allocation policies, plus the default policy WSRM policy named WSRMDefault, as shown in Figure 4-11.



```
Windows System Resource Manager (Local)
  Resource Allocation Policies
    Weekday
    Night
    Weekend
    WSRMDefault [Manage]
  Process Matching Criteria
    NewOrderEntry
    OrderTracking
    HRManagement
    BatchBilling
  Calendar [Disabled]
  Resource Monitor
  Accounting [Enabled]
```

Figure 4-11 New Resource Allocation Policies

4.4.5 Creating schedule and calendar entries

You can use the WSRM calendar to schedule the automatic activation of a resource allocation policy. You can specify that this be done on a one-time or recurring basis. You can also use a resource allocation policy schedule to manage multiple resource allocation policies and to activate different resource allocation policies at different, predefined times over a 24-hour period.

When the calendar is enabled, the default calendar resource allocation policy is in effect. You can configure the calendar default policy to be a user-defined resource allocation policy or use the default calendar resource allocation policy. If no specific policy is scheduled, WSRM applies the default calendar resource allocation policy. You can also configure the calendar to use a resource allocation policy other than the default policy for times when no other policies are scheduled.

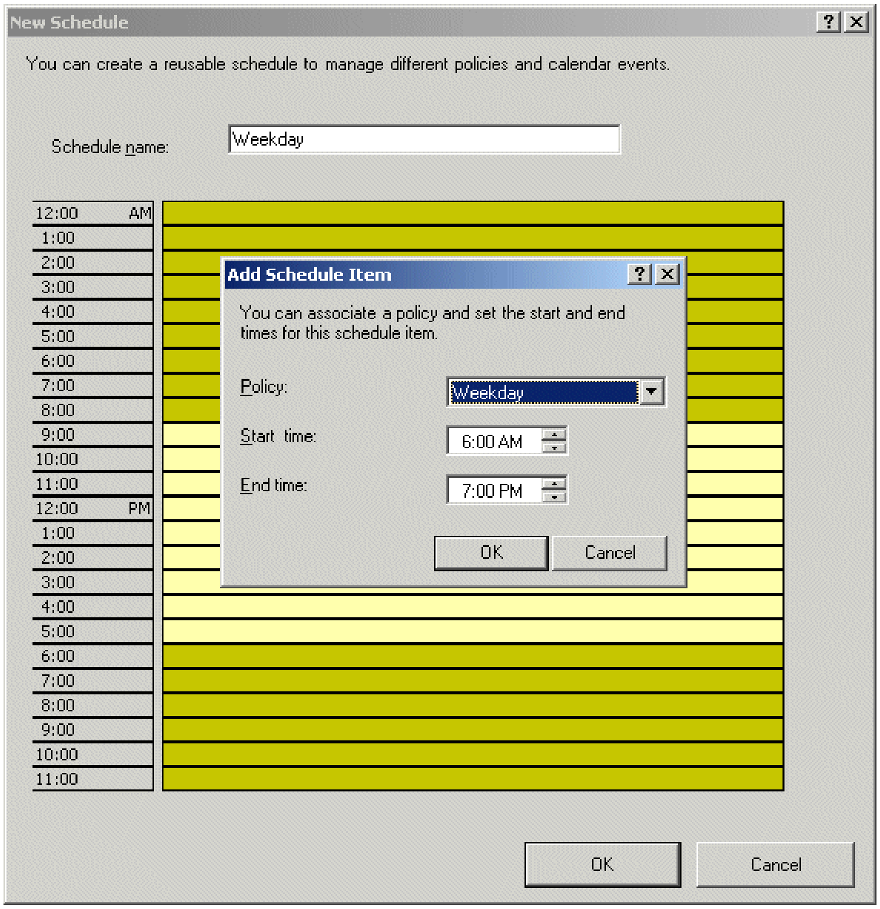
You can use the WSRM calendar to schedule the automatic activation of a resource allocation policy based on the date, the time, or both. In addition, you can manage multiple resource allocation policies by using a policy schedule. The schedule can activate different resource allocation policies at different, predefined times.

> **Note:** When a policy is manually selected to be the managing policy (that is, enabled other than through the calendar), that policy overrides the policy that was in effect by use of the WSRM calendar. Scheduling the enabling of resource allocation policies by means of the calendar might be the most effective way to meet business needs.

Creating a schedule

To create a new schedule, do the following:

1. Under Calendar in the WSRM console, right-click **Schedules** and select **New Schedule**.
2. Enter a new schedule name. We entered `Weekday`.
3. Double-click one of the timeslots to display the Add Schedule Item window. It does not matter which one you double-click, but the one you do use will be entered into the Start time field. See Figure 4-12.



New Schedule

You can create a reusable schedule to manage different policies and calendar events.

Schedule name: Weekday

Add Schedule Item

You can associate a policy and set the start and end times for this schedule item.

Policy: Weekday

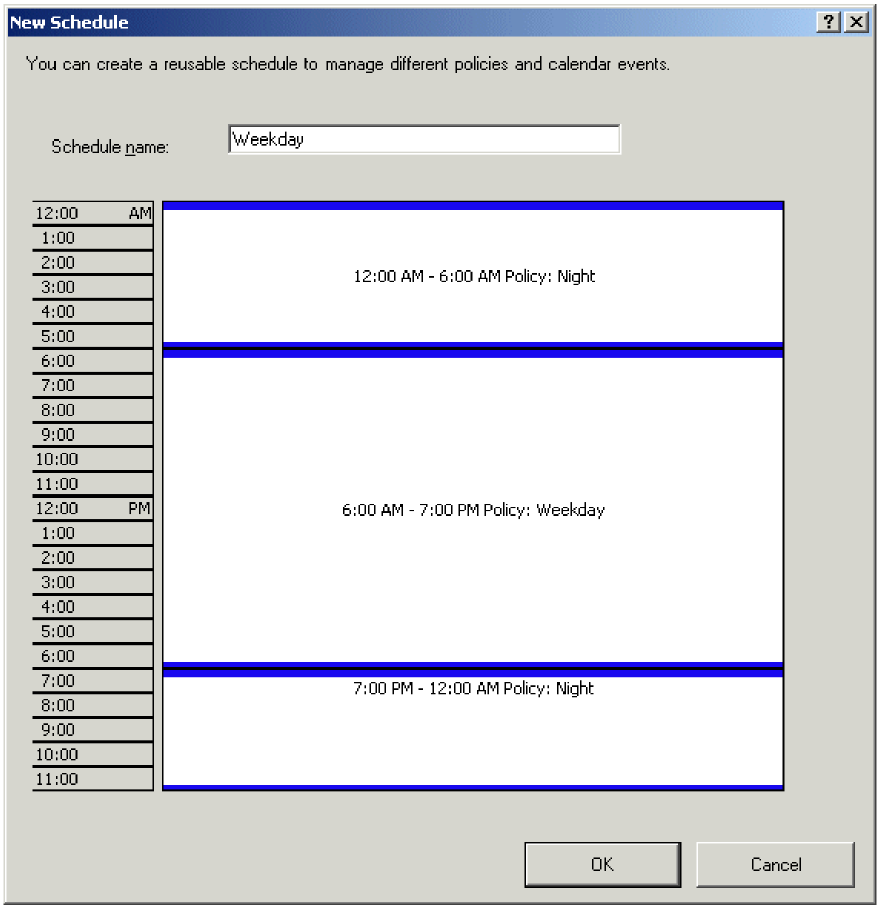
Start time: 6:00 AM

End time: 7:00 PM

Figure 4-12 Adding policies to a schedule

4. Select the policy you want to enable and the start and end times. We selected the Weekday policy we created earlier, and 5 a.m. to 7 p.m. as the times the policy should be in effect.
5. Click **OK**.
6. Double-click another time slot and enter the details for the next policy, Night, and specify the times to be 7 p.m. to 12 a.m. and then another one for 12 a.m. to 6 a.m. (that is, you need to create two calendar entries for this).

   The resulting schedule will look like the schedule shown in Figure 4-13.



New Schedule

You can create a reusable schedule to manage different policies and calendar events.

Schedule name: Weekday

12:00 AM - 6:00 AM Policy: Night

6:00 AM - 7:00 PM Policy: Weekday

7:00 PM - 12:00 AM Policy: Night

Figure 4-13 Multiple policies within one schedule

7. Click **OK** to save the schedule.

The WSRM management console now displays entries for each policy under Schedules.

Tip: The WSRM calendar is enabled by default. If the calendar is disabled, any scheduled policies will not run.

4.4.6 Creating calendar events

Now that you have created weekday and weekend schedules, the next step is to create a calendar that includes those schedules. WSRM uses the calendar to apply the correct resource management policies to your system.

For our example, we create two recurring calendar events.

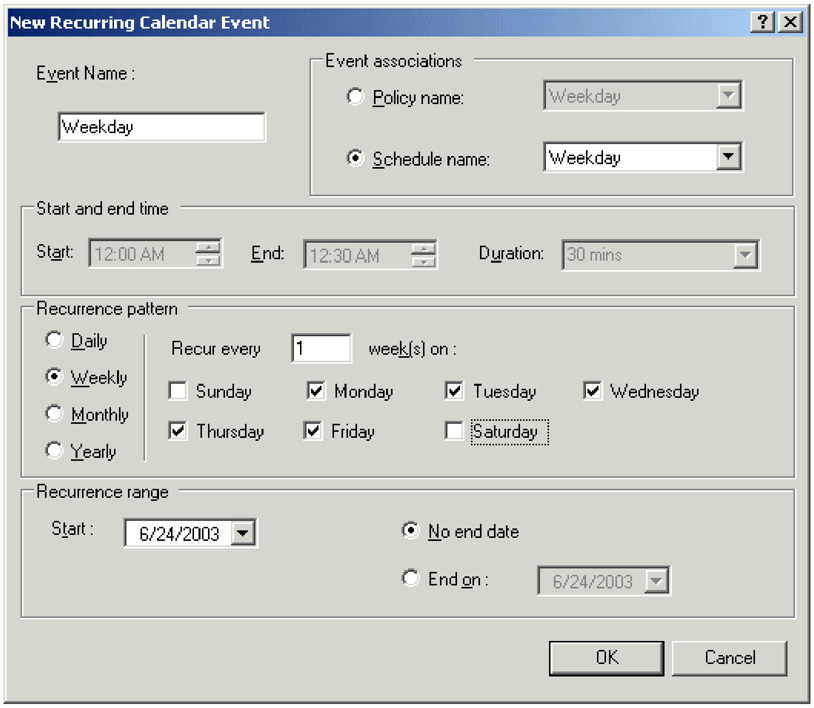
- One for the weekday schedule that contains weekday and night policies
- One for the weekend schedule that contains weekend and night policies

To create calendar events:

1. Right-click **Calendar Events** and select **New Recurring Calendar Event**.

2. Fill in the fields to set up the Weekday schedule as follows (see Figure 4-14):

   - Event Name: Type in `Weekday`.
   - Event associations: Select **Schedule name** and select **Weekday**.
   - Recurrence pattern: Select **Weekly**.
   - Recur every: **1** week
   - Select all weekdays: Monday through Friday.
   - Recurrence range Start: Enter today's date.
   - Select **No end date**.



New Recurring Calendar Event

Event Name : Weekday

Event associations
○ Policy name: Weekday
● Schedule name: Weekday

Start and end time
Start: 12:00 AM End: 12:30 AM Duration: 30 mins

Recurrence pattern
○ Daily
● Weekly
○ Monthly
○ Yearly

Recur every 1 week(s) on :
□ Sunday ☑ Monday ☑ Tuesday ☑ Wednesday
☑ Thursday ☑ Friday □ Saturday

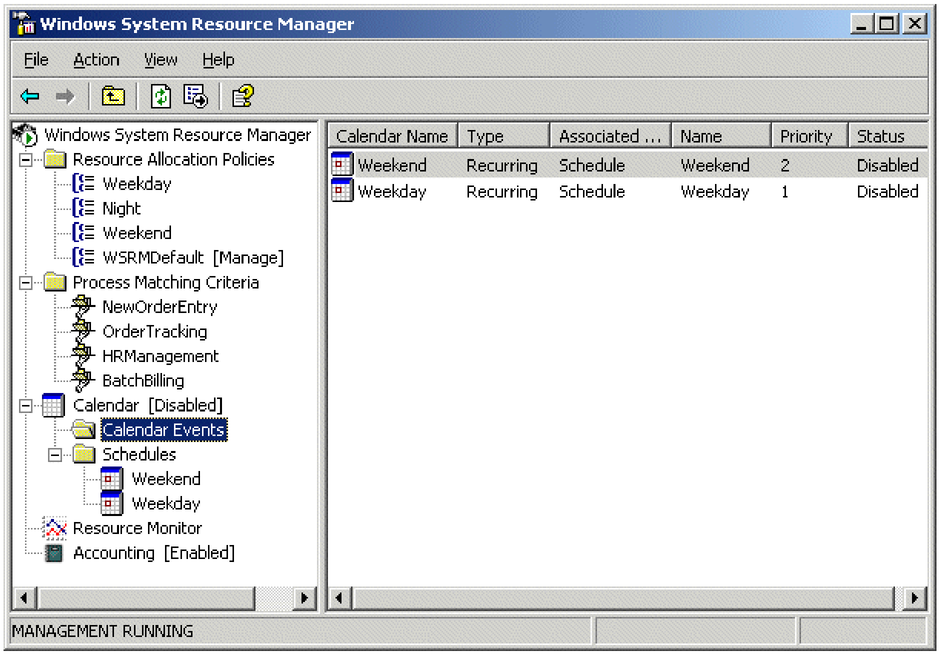
Recurrence range
Start : 6/24/2003
● No end date
○ End on : 6/24/2003

Figure 4-14 New Recurring Calendar Event window

3. Click **OK**.
4. Create a second recurring calendar event for the Weekend schedule:

   - Event Name: Type in `Weekend`.
   - Event associations: Select **Schedule name** and select **Weekday**.
   - Recurrence pattern: Select **Weekly**.
   - Recur every: **1** week.
   - Select Saturday and Sunday.
   - Recurrence range Start: Enter today's date.
   - Select **No end date**.

5. Click **OK**. The new calendar events are displayed in the console, as shown in Figure 4-15.



Calendar Name	Type	Associated ...	Name	Priority	Status
Weekend	Recurring	Schedule	Weekend	2	Disabled
Weekday	Recurring	Schedule	Weekday	1	Disabled

Figure 4-15 New Calendar Events view

4.4.7 Activating WSRM resource management

As shown in Figure 4-15, the calendar is currently disabled (with [Disabled] next to it in the console) and the managing policy is the default WSRM policy, WSRMDefault. To activate the schedule-based policies that you defined, enable the calendar by right-clicking **Calendar** and selecting **Enable**. The policy that should be active based on the calendar definitions should now be marked with [Manage] next to it.

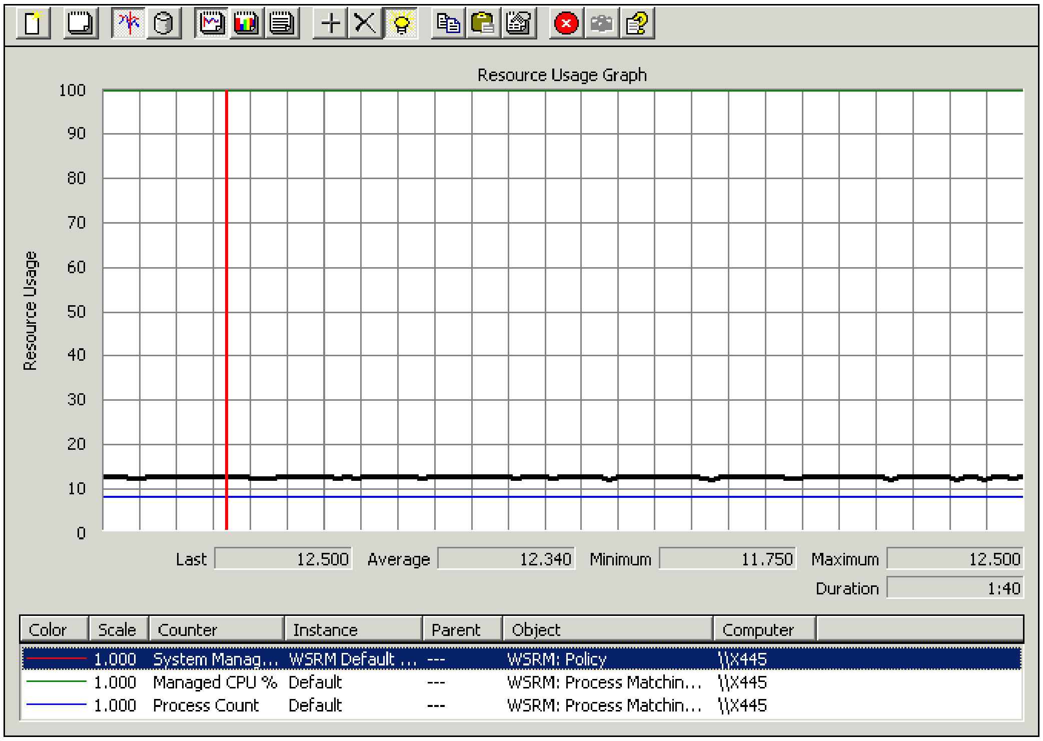
WSRM is now active. The calendar notified WSRM to automatically start the correct policy. The correct resource allocation policies decision was based on the information you had already provided from the process matching criteria, resource allocation entries, schedules, and calendar events.

4.4.8 Monitoring performance

The Resource Monitor lets you monitor the performance levels of your server using Windows Server 2003 System Monitor without having to leave the WSRM console. As described in 1.3.4, “Resource Monitor” on page 7, there are three additional performance objects that WSRM provides:

- WSRM policy
- WSRM process
- WSRM process matching criteria

Select **Resource Monitor** from the left pane of the WSRM management console and the Resource Usage Graph displays the default counters, as shown in Figure 4-16.



Last 12.500 Average 12.340 Minimum 11.750 Maximum 12.500
Duration 1:40

Color	Scale	Counter	Instance	Parent	Object	Computer
🟥	1.000	System Manag...	WSRM Default ...	---	WSRM: Policy	\\X445
🟩	1.000	Managed CPU %	Default	---	WSRM: Process Matchin...	\\X445
🟦	1.000	Process Count	Default	---	WSRM: Process Matchin...	\\X445

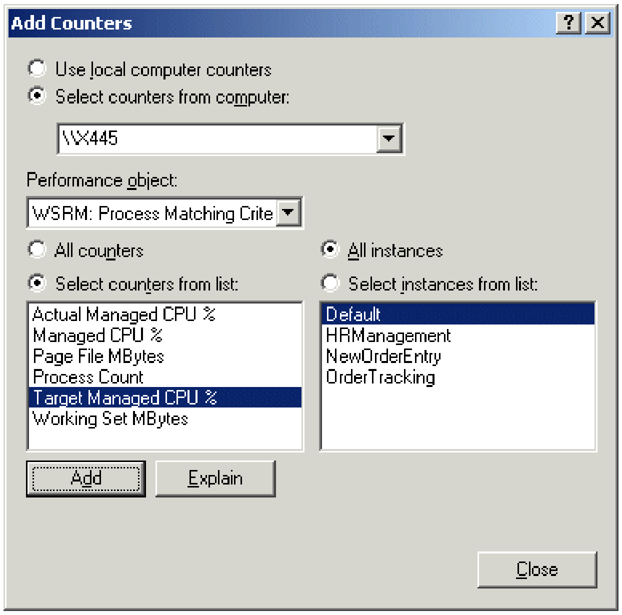
Figure 4-16 Resource Monitor with the default counters shown

If you set a resource allocation policy to managing status, Resource Monitor automatically populates some counters within the Resource Monitor.

For our example, delete these existing counters. This keeps the display less cluttered and allows you to better understand what WSRM is doing.

With Resource Monitor, you can view a graph of your target (or goal) CPU settings for each instance (application) and the actual CPU usage for each instance. This is set up as follows:

1. Right-click anywhere on the graph and select **Add Counters**, or click the + button. The Add Counters window opens, as shown in Figure 4-17.



Add Counters

○ Use local computer counters
● Select counters from computer:

\\X445

Performance object:

WSRM: Process Matching Crite

○ All counters
● Select counters from list:

Actual Managed CPU %
Managed CPU %
Page File MBytes
Process Count
Target Managed CPU %
Working Set MBytes

● All instances
○ Select instances from list:

Default
HRManagement
NewOrderEntry
OrderTracking

Figure 4-17 Adding counters to Resource Monitor

2. For Performance object, select **WSRM: Process Matching Criteria**.
3. For Select counters from list, select **Target Managed CPU%**.
4. Select **All instances**.

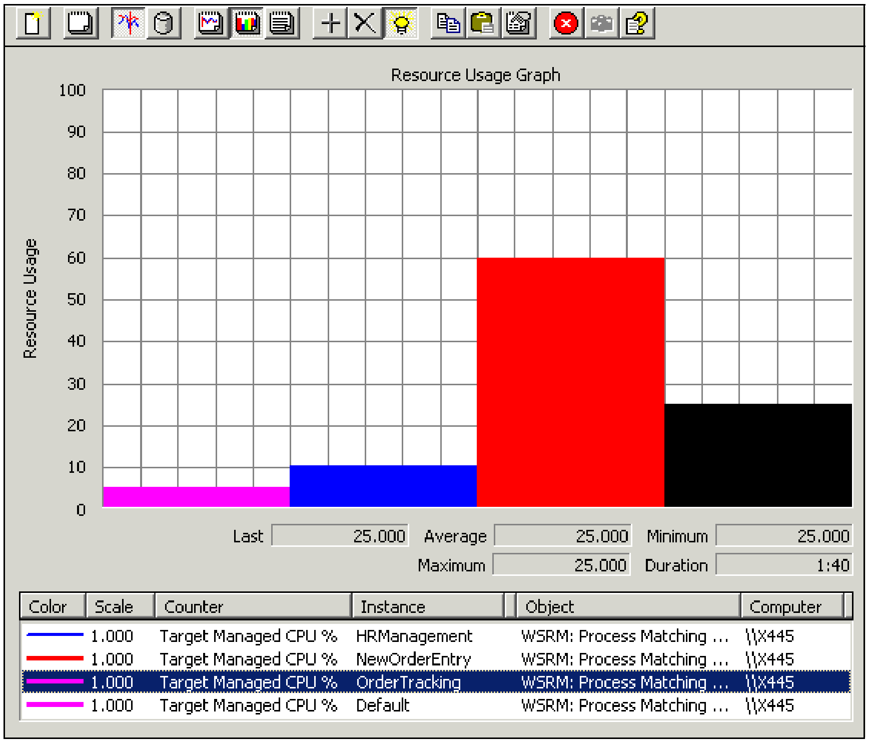
   If your list of instances does not match the list in Figure 4-17, it is likely the Weekday resource allocation policy is not set to be the Managing policy or is not set up correctly.

   > **Note:** BatchBilling is not present because it is only active when the Night schedule is active.

5. Click **Close**.

After you close the Add Counters window, Resource Monitor displays your target CPU% settings that you specified, as shown in Figure 4-18. This is not your actual usage, just your targets. It should match the Server Policy table, as shown in Table 4-1 on page 29.

- NewOrderEntry is set to 60% managed CPU.
- OrderTracking is set to 25% managed CPU.
- HRManagement is set to 10% managed CPU.
- Default is set to the remainder, or 5% managed CPU.



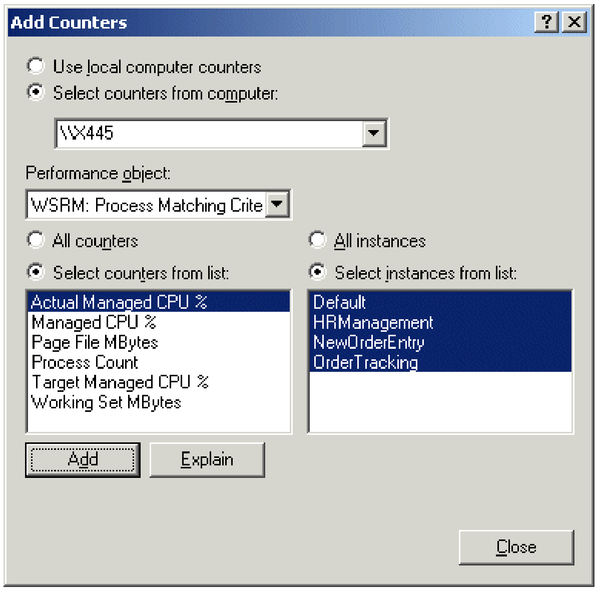
Last 25.000 Average 25.000 Minimum 25.000
Maximum 25.000 Duration 1:40

Color	Scale	Counter	Instance	Object	Computer
🟦	1.000	Target Managed CPU %	HRManagement	WSRM: Process Matching ...	\\X445
🟥	1.000	Target Managed CPU %	NewOrderEntry	WSRM: Process Matching ...	\\X445
🟪	1.000	Target Managed CPU %	OrderTracking	WSRM: Process Matching ...	\\X445
🟪	1.000	Target Managed CPU %	Default	WSRM: Process Matching ...	\\X445

Figure 4-18 Target CPU% displayed as a histogram

To see the actual performance of applications running under the policy, use the Actual Managed CPU% counter:

1. Add a new counter.
2. Select the performance object **WSRM Performance Matching Criteria**.
3. Select the counter **Actual Managed CPU %**.
4. Select the same set of instances as was earlier done for the Target Managed CPU. See Figure 4-19.



○ Use local computer counters
● Select counters from computer:

\\X445

Performance object:

WSRM: Process Matching Crite

○ All counters
● Select counters from list:

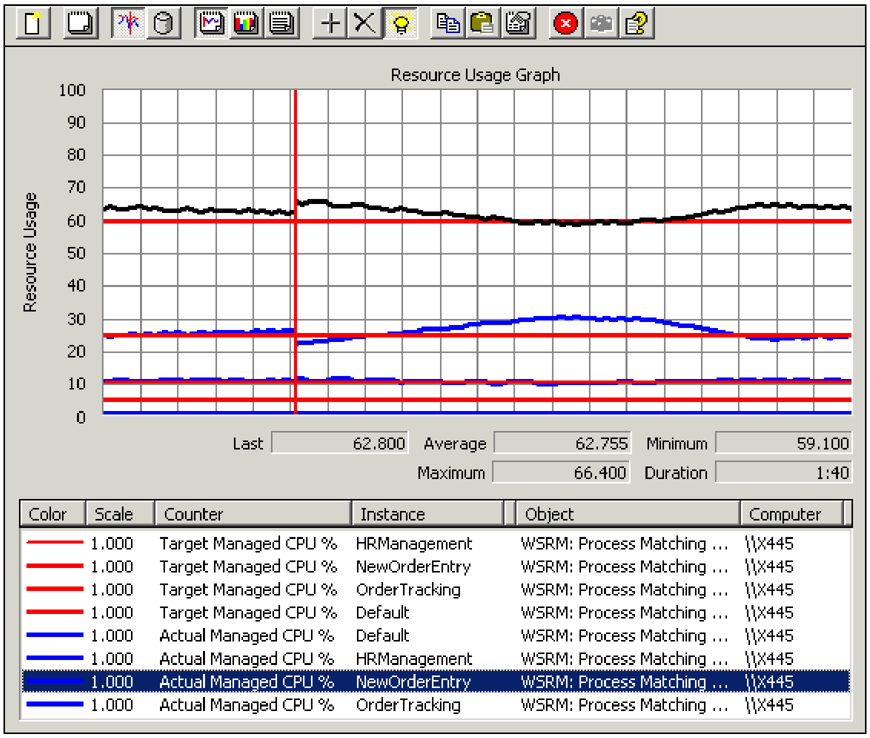
- Actual Managed CPU % (selected)
- Managed CPU %
- Page File MBytes
- Process Count
- Target Managed CPU %
- Working Set MBytes

○ All instances
● Select instances from list:

- Default (selected)
- HRManagement (selected)
- NewOrderEntry (selected)
- OrderTracking (selected)

Figure 4-19 Actual usage counters

You should now see how your system is managing its CPU resources in contrast to what you planned or targeted. In Figure 4-20, note how the actual usage moves above and below the specified target as resources are reallocated.



Last 62.800 Average 62.755 Minimum 59.100
Maximum 66.400 Duration 1:40

Color	Scale	Counter	Instance	Object	Computer
🟥	1.000	Target Managed CPU %	HRManagement	WSRM: Process Matching ...	\\X445
🟥	1.000	Target Managed CPU %	NewOrderEntry	WSRM: Process Matching ...	\\X445
🟥	1.000	Target Managed CPU %	OrderTracking	WSRM: Process Matching ...	\\X445
🟥	1.000	Target Managed CPU %	Default	WSRM: Process Matching ...	\\X445
🟦	1.000	Actual Managed CPU %	Default	WSRM: Process Matching ...	\\X445
🟦	1.000	Actual Managed CPU %	HRManagement	WSRM: Process Matching ...	\\X445
🟦	1.000	Actual Managed CPU %	NewOrderEntry	WSRM: Process Matching ...	\\X445
🟦	1.000	Actual Managed CPU %	OrderTracking	WSRM: Process Matching ...	\\X445

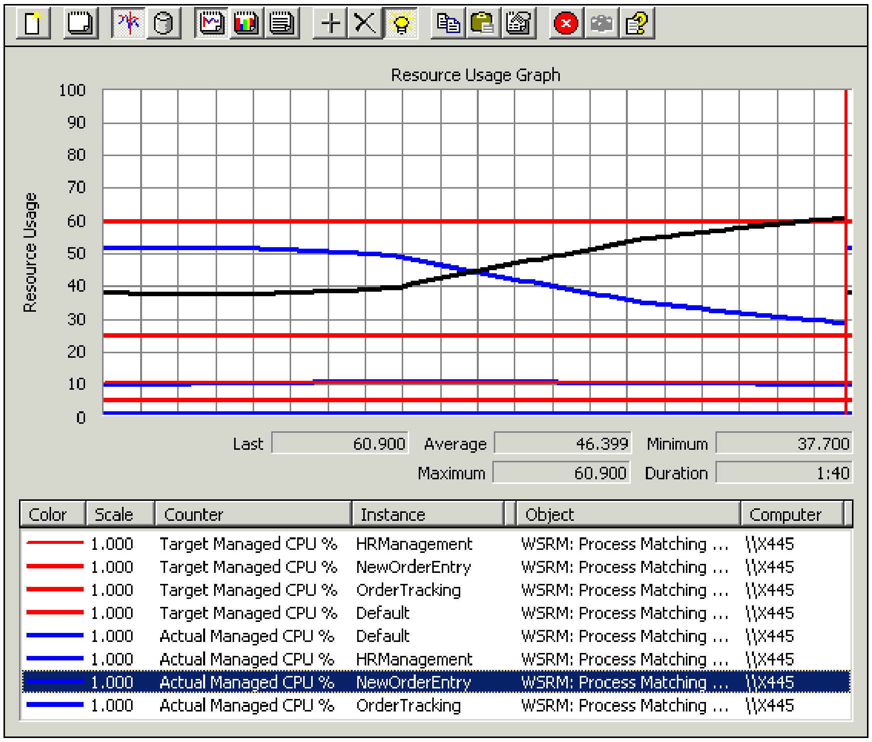
Figure 4-20 Actual CPU usage versus target usage

After the applications are running and contending for system CPU resources, WSRM allocates resources as planned. If an application is not demanding resources, WSRM reallocates available CPU resources to demanding applications. This efficient use of CPU resources allows for applications to get more resources when needed.

4.5 How WSRM handles resource contention

As the NewOrderEntry program activity reaches its peak activity hours, WSRM can reallocate more CPU resources to it, even beyond its allotted 60%. When other applications are not demanding CPU resources, they are diverted to NewOrderEntry.

In Figure 4-21 on page 45, as NewOrderEntry is gaining CPU resources, OrderTracking is seen declining from 52% to 30% Actual Managed CPU. Although OrderTracking is probably demanding more CPU resources, the resources are allocated away from it, because WSRM has defined NewOrderEntry to receive 60%. If NewOrderEntry CPU demands push beyond 60%, OrderTracking is still guaranteed 25%. Any additional resources for either application comes from the default group or any managed application currently or momentarily not demanding CPU resources.

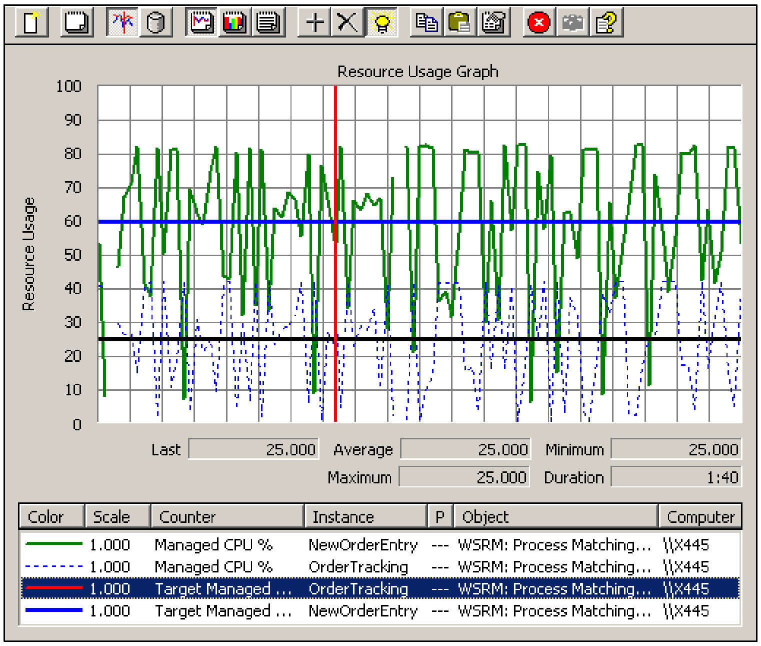


Last 60.900 Average 46.399 Minimum 37.700
Maximum 60.900 Duration 1:40

Color	Scale	Counter	Instance	Object	Computer
🟥	1.000	Target Managed CPU %	HRManagement	WSRM: Process Matching ...	\\X445
🟥	1.000	Target Managed CPU %	NewOrderEntry	WSRM: Process Matching ...	\\X445
🟥	1.000	Target Managed CPU %	OrderTracking	WSRM: Process Matching ...	\\X445
🟥	1.000	Target Managed CPU %	Default	WSRM: Process Matching ...	\\X445
🟦	1.000	Actual Managed CPU %	Default	WSRM: Process Matching ...	\\X445
🟦	1.000	Actual Managed CPU %	HRManagement	WSRM: Process Matching ...	\\X445
🟦	1.000	Actual Managed CPU %	NewOrderEntry	WSRM: Process Matching ...	\\X445
🟦	1.000	Actual Managed CPU %	OrderTracking	WSRM: Process Matching ...	\\X445

Figure 4-21 The NewOrderEntry program begins reaching a peak usage period

Figure 4-22 shows both applications stressed by client requests. Their designated target CPU% is set at 60% for NewOrderEntry and 25% for OrderTracking. Yet WSRM is able to provide each application with additional needed resources. This is observed in the spikes that rise and fall around each target CPU% line.



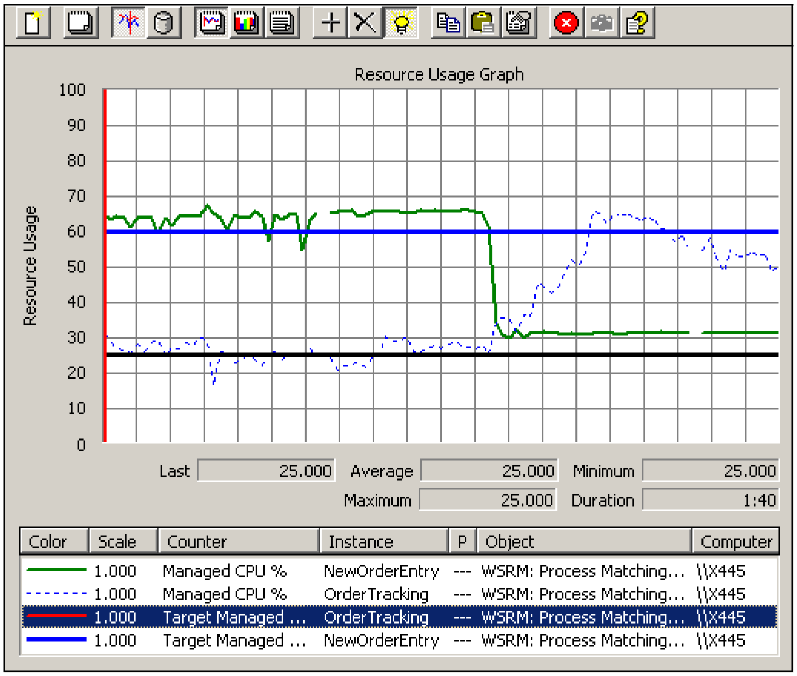
Last 25.000 Average 25.000 Minimum 25.000
Maximum 25.000 Duration 1:40

Color	Scale	Counter	Instance	P	Object	Computer
🟩	1.000	Managed CPU %	NewOrderEntry	---	WSRM: Process Matching...	\\X445
┄	1.000	Managed CPU %	OrderTracking	---	WSRM: Process Matching...	\\X445
🟥	1.000	Target Managed ...	OrderTracking	---	WSRM: Process Matching...	\\X445
🟦	1.000	Target Managed ...	NewOrderEntry	---	WSRM: Process Matching...	\\X445

Figure 4-22 Graph of heavy usage period for both applications

In Figure 4-23, a dramatic decline is seen in usage of the NewOrderEntry application that was allotted 60% target CPU resources. The OrderTracking program that was given only 25%

CPU target resources had been demanding more CPU resources, but until now could not gain them. As client access declines for NewOrderEntry, CPU resources are automatically reallocated by WSRM. OrderTracking is able to acquire unused CPU resources. However, if at any time the NewOrderEntry program requires more CPU, and has to compete with the OrderTracking program, WSRM moves to the defined parameters and ensures that OrderEntry can get at least 60% of the CPU resources.



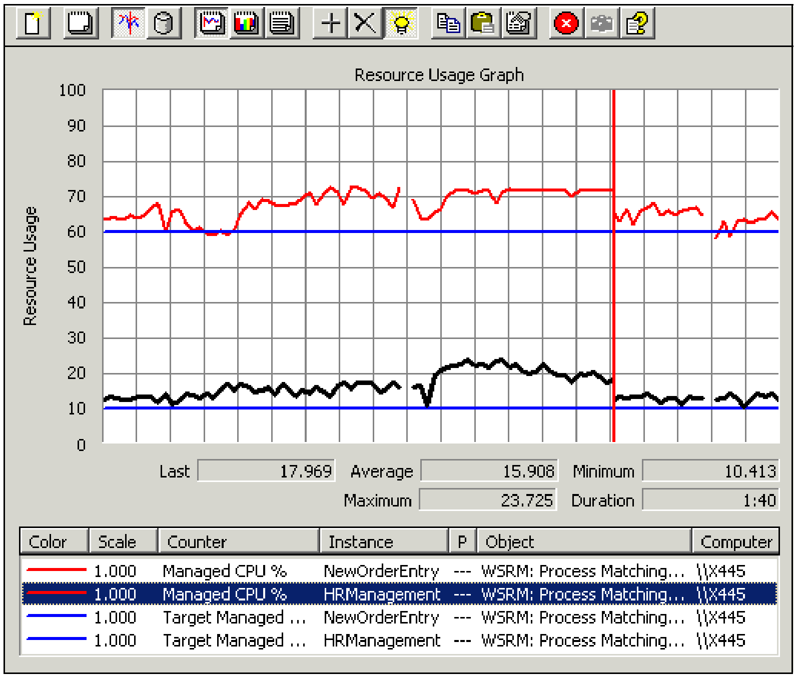
Last 25.000 Average 25.000 Minimum 25.000
Maximum 25.000 Duration 1:40

Color	Scale	Counter	Instance	P	Object	Computer
🟩	1.000	Managed CPU %	NewOrderEntry	---	WSRM: Process Matching...	\\X445
╌	1.000	Managed CPU %	OrderTracking	---	WSRM: Process Matching...	\\X445
🟥	1.000	Target Managed ...	OrderTracking	---	WSRM: Process Matching...	\\X445
🟦	1.000	Target Managed ...	NewOrderEntry	---	WSRM: Process Matching...	\\X445

Figure 4-23 WSRM reallocates resources to OrderTracking

Human Resources had contributed 10% of the cost of the server. In return, they received a guarantee of 10% of the system's CPU resources. Even during a peak period for NewOrderEntry, the Human Resources application, HRM.EXE, still receives 10% of the resources. This is because all resource allocation policies define HRManagement to receive 10% CPU resources.

In Figure 4-24 on page 47, how is it that both applications shown are receiving *over* their allotted resources at the same time? Because OrderTracking (25% CPU target) and the default group programs are not being heavily accessed, WSRM has additional resources available to provide to NewOrderEntry and HRManagement.



Last	17.969	Average	15.908	Minimum	10.413
		Maximum	23.725	Duration	1:40

Color	Scale	Counter	Instance	P	Object	Computer
🟥	1.000	Managed CPU %	NewOrderEntry	---	WSRM: Process Matching...	\\X445
🟥	1.000	Managed CPU %	HRManagement	---	WSRM: Process Matching...	\\X445
🟦	1.000	Target Managed ...	NewOrderEntry	---	WSRM: Process Matching...	\\X445
🟦	1.000	Target Managed ...	HRManagement	---	WSRM: Process Matching...	\\X445

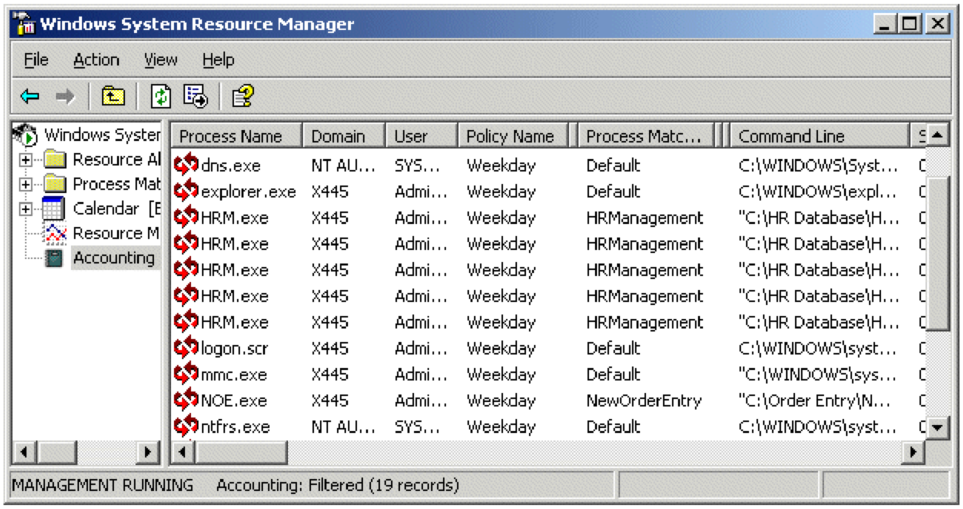
Figure 4-24 WSRM guarantees 10% CPU resources to the Human Resources application

4.6 Accounting

Accounting data is provided to Human Resources because they were guaranteed 10% of the server's resources.

WSRM stores accounting information on resource utilization. Reports can be generated to show how resources were allocated. This can be used in charge-back accounting. In additional, you can save snapshots of Resource Monitoring in HTM/HTML and show resource allocation policies with designated resource allocations.

Right-click **Accounting** and select **Filter View**. You can customize reports to show specific fields, processes, and statistics, as shown in Figure 4-25.



Process Name	Domain	User	Policy Name	Process Matc...	Command Line	S
dns.exe	NT AU...	SYS...	Weekday	Default	C:\WINDOWS\Syst...	C
explorer.exe	X445	Admi...	Weekday	Default	C:\WINDOWS\expl...	C
HRM.exe	X445	Admi...	Weekday	HRManagement	"C:\HR Database\H...	C
HRM.exe	X445	Admi...	Weekday	HRManagement	"C:\HR Database\H...	C
HRM.exe	X445	Admi...	Weekday	HRManagement	"C:\HR Database\H...	C
HRM.exe	X445	Admi...	Weekday	HRManagement	"C:\HR Database\H...	C
HRM.exe	X445	Admi...	Weekday	HRManagement	"C:\HR Database\H...	C
logon.scr	X445	Admi...	Weekday	Default	C:\WINDOWS\syst...	C
mmc.exe	X445	Admi...	Weekday	Default	"C:\WINDOWS\sys...	C
NOE.exe	X445	Admi...	Weekday	NewOrderEntry	"C:\Order Entry\N...	C
ntfrs.exe	NT AU...	SYS...	Weekday	Default	C:\WINDOWS\syst...	C

MANAGEMENT RUNNING   Accounting: Filtered (19 records)

Figure 4-25 Accounting view

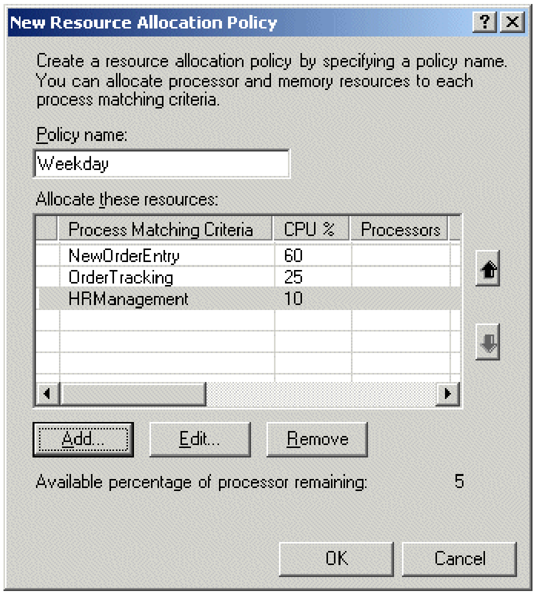
4.7 CPU affinity and memory resource assignment

Managing CPU% resource allocations is reasonably simple, and the results are somewhat predictable. With WSRM, you can also manage processor affinity and work with both committed and working set memory. The impact to performance is more difficult to predict. There is a distinct possibility that these settings could just as easily degrade performance as improve them. If you are going to use memory resource settings, be sure you have a measurable way to determine if the impact of the changes are positive.

4.7.1 CPU affinity

Let us say you are using an IBM eServer xSeries 445 Server with eight CPUs, and you want the NewOrderEntry application to use all four processors on one SMP Expansion Module, and the other two applications to share the four processors of the other SMP Expansion Module. It was decided that during the normal business hours, NewOrderEntry would run on four CPUs, OrderTracking would run on two CPUs, and HRManagement would run on one CPU.

In the left pane of the WSRM management console, right-click the **Weekday** resource allocation policy and select **Properties**. It shows the original settings, as shown in Figure 4-26.



Create a resource allocation policy by specifying a policy name. You can allocate processor and memory resources to each process matching criteria.

Policy name:
Weekday

Allocate these resources:

Process Matching Criteria	CPU %	Processors
NewOrderEntry	60	
OrderTracking	25	
HRManagement	10	

Available percentage of processor remaining: 5

Figure 4-26 The Weekday resource allocation policy

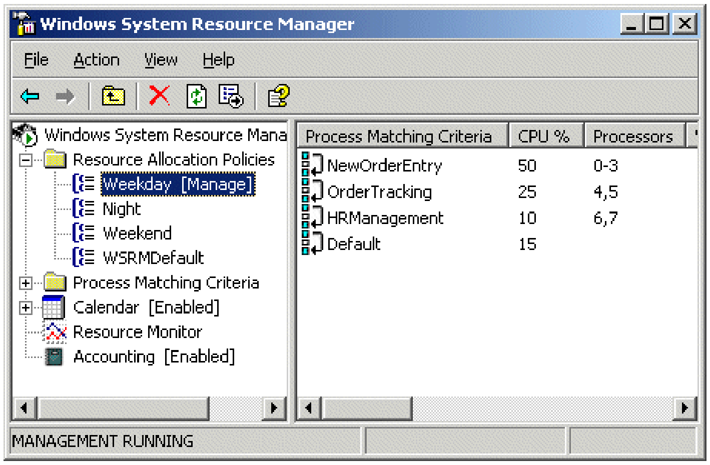
When you use processor affinity, the number entered in the CPU% is calculated differently. For example, let us say we are limiting NewOrderEntry to receive the first four of eight (4/8 or 50%) CPUs in the system. It, therefore, would only get 50% of the total system's CPU resources. We must first edit the NewOrderEntry CPU% value to 50% CPU before we can assign this application processors 0 through 3.

You need to select each one of these three process criteria one at a time:

1. Select **NewOrderEntry** and click **Edit**. On the General tab, change the Percentage of processor allocated from 60% to 50%, our maximum allowed for four CPUs. Select the Advanced tab. Select the **Use Specified Processors** check box, and enter `0-3`. Click **OK**.

2. Select **OrderTracking** and click **Edit**. On the General tab, leave the Percentage of processor allocated at 25%. Select the Advanced tab. Select the **Use Specified Processors** check box, and enter `4,5`. Click **OK**.
3. Select **HRManagement** and click **Edit**. On the General tab, leave the Percentage of processor allocated at 10%. Select the Advanced tab. Select the **Use Specified Processors** check box, and enter `6,7`. Click **OK**.

When you select the Weekday resource allocation policy, you will now see CPU% followed by selected processors (processor affinity), as shown in Figure 4-27.

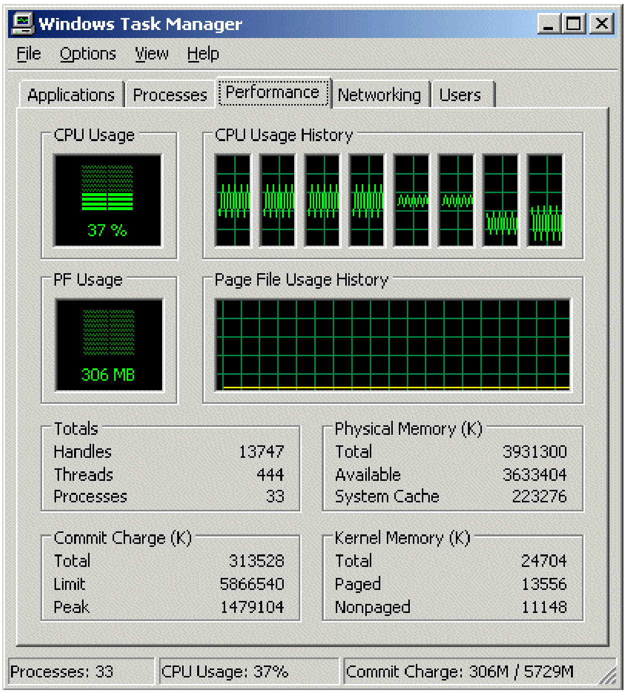


Process Matching Criteria	CPU %	Processors
NewOrderEntry	50	0-3
OrderTracking	25	4,5
HRManagement	10	6,7
Default	15	

Figure 4-27 Processor affinity shown for a resource allocation policy

You do not have to restrict one application to a group of CPUs and another application to a different group as we did above. You can also have one application with access to all of the CPUs, while another application is limited to fewer CPUs.

If you display the Windows Task Manager, as shown in Figure 4-28, you will see managed resources limited to specific CPUs.



Totals
Handles 13747
Threads 444
Processes 33

Physical Memory (K)
Total 3931300
Available 3633404
System Cache 223276

Commit Charge (K)
Total 313528
Limit 5866540
Peak 1479104

Kernel Memory (K)
Total 24704
Paged 13556
Nonpaged 11148

Processes: 33 CPU Usage: 37% Commit Charge: 306M / 5729M

Figure 4-28 Task Manager view of CPU utilization for affinitized CPUs

4.7.2 Memory resource allocation

Let us say you needed to run a new application on your server named rmconsumer.exe. However, the application was known to either have a memory leak or consume too much memory. You decided to limit its memory resources.

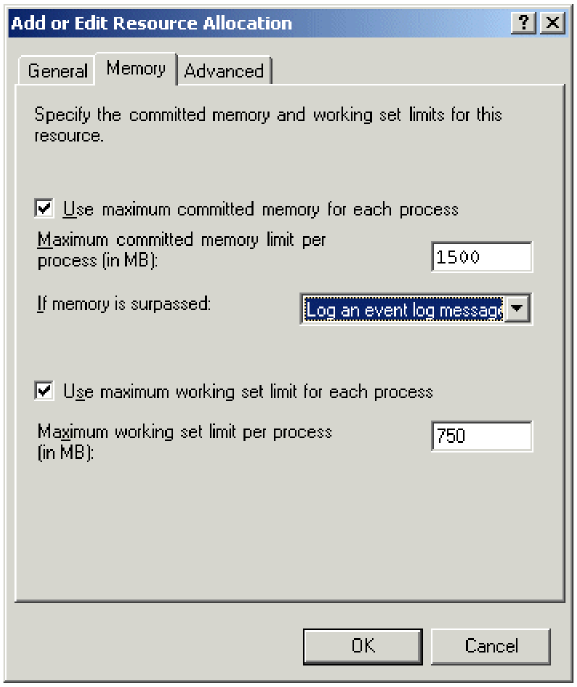
When rmconsumer.exe exceeds its memory limit, WSRM can stop the application or log an entry in the event log, or both. You decided the application should be allowed to continue running, but want to receive an event log notification.

First, create a process matching criteria named Consumer:

1. Right-click **Process Matching Criteria** and select **New Process Matching Criteria**.
2. Browse and select your application, c:\consumer\rmconsumer.exe.

Add the new criteria named Consumer to the Weekday resource allocation policy:

1. Right-click **Resource Allocation Policy** and select **Properties**.
2. On the General tab, select **Consumer** from the drop-down menu and assign it CPU resources.
3. On the Memory tab, select **Use maximum committed memory**. Enter `1500`. Also select **Log an event** from the drop-down menu. Select **Use maximum working set** and enter `750`.



Add or Edit Resource Allocation

Specify the committed memory and working set limits for this resource.

☑ Use maximum committed memory for each process

Maximum committed memory limit per process (in MB): 1500

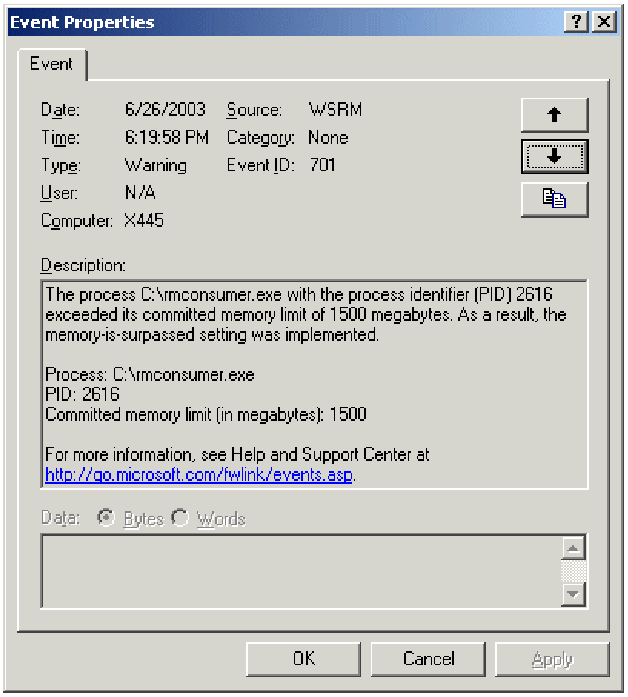
If memory is surpassed: Log an event log message

☑ Use maximum working set limit for each process

Maximum working set limit per process (in MB): 750

Figure 4-29 Entering memory resource allocation limits

If rmconsumer.exe does exceed its limits, the event shown in Figure 4-30 is logged.



Event Properties

Date: 6/26/2003 Source: WSRM
Time: 6:19:58 PM Category: None
Type: Warning Event ID: 701
User: N/A
Computer: X445

Description:

The process C:\rmconsumer.exe with the process identifier (PID) 2616 exceeded its committed memory limit of 1500 megabytes. As a result, the memory-is-surpassed setting was implemented.

Process: C:\rmconsumer.exe
PID: 2616
Committed memory limit (in megabytes): 1500

For more information, see Help and Support Center at http://go.microsoft.com/fwlink/events.asp.

Data: ● Bytes ○ Words

Figure 4-30 Event log entry indicating memory allocation was exceeded

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this Redpaper.

IBM Redbooks

For information on ordering these publications, see "How to get IBM Redbooks" on page 53. Note that some of the documents referenced here may be available in softcopy only.

- *IBM @server xSeries 445 Planning and Installation Guide*, SG24-8870
- *Implementing Systems Management Solutions using IBM Director*, SG24-6188
- *Windows Server 2003, Datacenter Edition on the IBM @server xSeries 445*, REDP-3700

Online resources

These Web sites and URLs are also relevant as further information sources:

- Microsoft Windows System Resource Manager FAQ

  `http://www.microsoft.com/windowsserver2003/downloads/wsrmfaq.mspx`
- Downloading Windows System Resource Manager

  `http://www.microsoft.com/windowsserver2003/downloads/wsrm.mspx`
- IBM Director Application Workload Manager

  `http://www.ibm.com/servers/eserver/xseries/systems_management/director_4/awm.html`
- Hyper-Threading Technology

  `http://www.intel.com/technology/hyperthread`

How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

IBM®

Implementing Windows System Resource Manager

Redpaper

Describes how WSRM helps you manage CPU and memory resources

Includes positioning and planning information

Provides information about how to configure WSRM

This Redpaper introduces Microsoft Windows System Resource Manager (WSRM), a component of Windows Server 2003, Datacenter Edition and Enterprise Edition. WSRM is a utility used for managing CPU and memory resource allocation among multiple applications that run on the same operating system image.

This paper describes what the tool does and how it works. It also describes how to prepare to use WSRM and how to implement a basic configuration of WSRM.

The paper will help Windows Server 2003 customers, especially those implementing an IBM Eserver xSeries 445 solution, and those people who want to understand how CPU and memory resource usage can be managed.