IBM Process Control Usage Guide

Leverage your servers with IBM Process Control

by Joakim Hansson and Adam Swellum
IBM Server Group

2. Abstract

IBM has given traditional system management strategies a new meaning in eLiza, self-managing servers. eLiza combines tools and techniques from various management strategies such as system management, workload management and application management into one overall management strategy.

eLiza extends the IBM @server platform to help:

- Provide resource management and automation capabilities
- Configure your system on the fly
- Repair problems online
- Defend against unauthorized access

A single point of control for all aspects of management involving a system or group of systems is one of the key points in the IBM system, workload and application management strategies for all Intel® processor-based systems.

Process Control, as part of the IBM workload and application management strategy, places emphasis on applications and their resource utilization on IBM @server xSeries severs.

The intent of this document is to provide Process Control “Best Practice” documentation and Process Control usage scenarios to enable organizations to increase their knowledge of Process Control and it’s many uses. For more information regarding Process Control please see the Process Control Help Files.

Please see the IBM web site for more information regarding eLiza and how IBM workload management and system management strategies can help your organization in the future.

3. Introduction to Process Control

Process Control is a powerful tool for organizing and managing processes and system resources on Intel processor-based Microsoft® Windows® 2000 Server systems, resulting in improved system throughput and application performance.

Process Control was developed by IBM and provided to Microsoft for inclusion in all Windows 2000 Datacenter implementations. Process Control on Windows 2000 Server and Advanced Server is an IBM exclusive and will be delivered on ftp://ftp.pc.ibm.com/pub/special/sysmgmt/ProcConPackage.exe

Organizations can deploy Process Control to solve the following types of problems:

- Differentiate two or more processes with the same image name
- Server consolidation by the use of CPU affinity and memory constraints
- Secure servers from unauthorized applications or processes
- Resource utilization, reporting and billing support

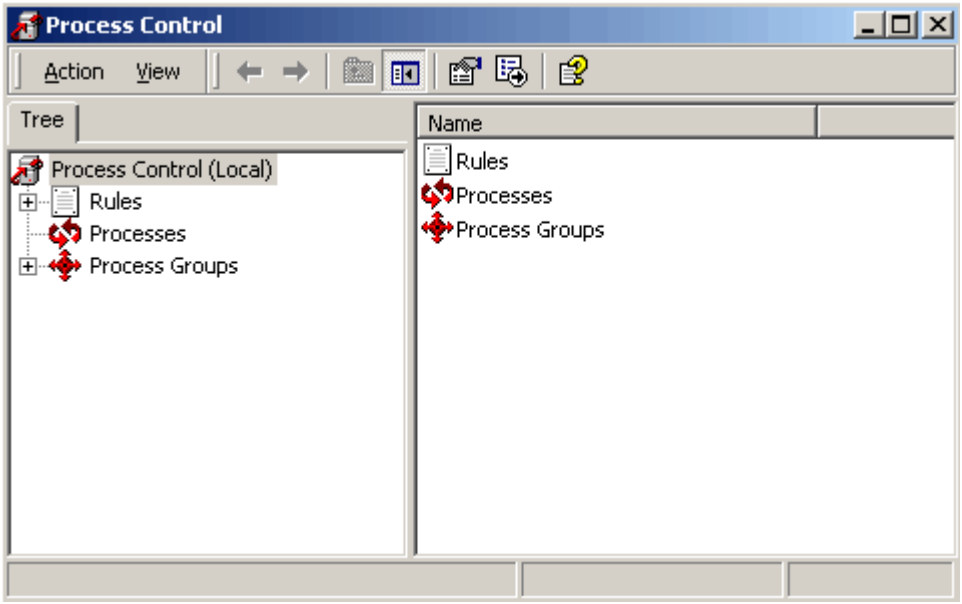
Organizations can control local or remote server processes and groups of processes with the easy to use Microsoft Management Console (MMC) or the built in Process Control Command Line Interface. Process Control was designed to complement, not replace, Windows 2000 Task Manager and System Monitor.

Process Control has the following requirements:

- IBM Intel processor based Servers
- Microsoft Windows 2000 Server, Advanced Server or Windows 2000 Datacenter
- Administrator privileges on the target server

4. Process Control Technical Overview

The core of Process Control is the usage of rules. Rules define the operation of Process Control when managing server systems.



There are three rule types in Process Control:

- Process Alias Rules
- Process Execution Rules
- Process Group Execution Rules

Process Alias Rules

Windows 2000 use a number (Process ID) tied to the process “Image Name” to identify a process. Process IDs change with each invocation of a process and may be reused by the system. The use of Image Name and Process IDs can cause identification problems in applications such as Task Manager when multiple processes have the same image name. It is however, possible to differentiate two or more processes with the same image name by defining a process alias rule.

Alias rules is the basic procedure required for managing processes.

Defining an alias in Process Control consists of two phases:

- Identifying a process or group of processes
- Giving the process or group of processes an alias name

Identifying a process or group of processes:

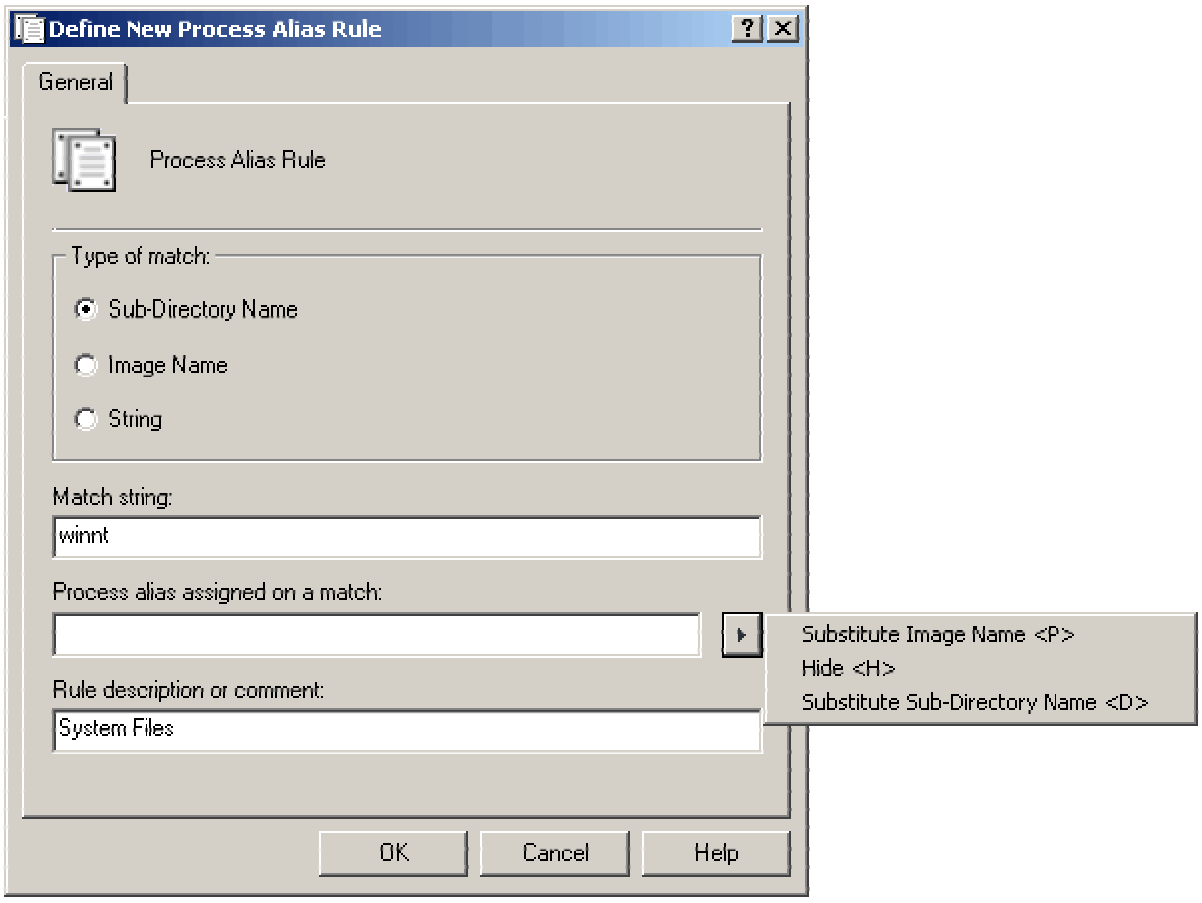
Process Control identifies one or many processes based on one of three character matching techniques.

- Sub Directory Name identifies processes based on a directory match
- Image Name identifies processes based on the process image name
- String Name identifies processes based on image names, directory matches or a combination of an image name and a directory

Wild cards and environment variables can be used in match strings (*) for all remaining characters and (?) for any character in this position.

Giving the process or group of processes an alias name:

User defined names for identified processes can consist of any characters except backslash (\), comma (,), or double quotation mark (“).

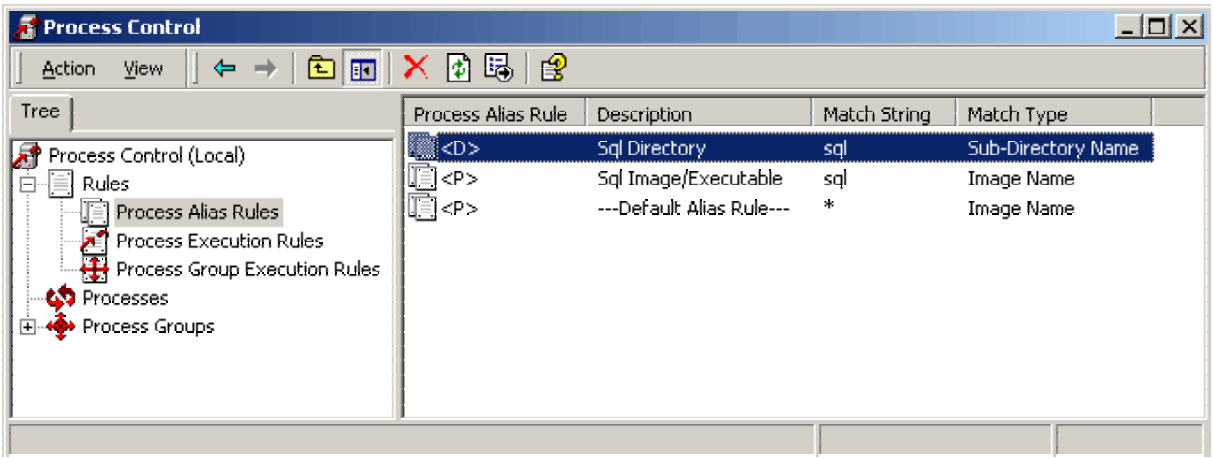


Process Control can furthermore assign names to aliases by the use of option tags. For example:

- The <P> tag assigns the image name of the executable as the alias
- The <D> tag assigns the directory name in which the process/executable started as the alias
- The <H> tag prevents identified processes from appearing in the process list

Process Control uses top-down precedence for a list of defined alias rules when assigning an alias name to a process. Identified processes will be given a defined alias name. By default all remaining processes are identified in the last alias rule, which is the default alias rule.

It is possible to change the precedence order of any given rule by right clicking the selected rule and then clicking on **Move up** or **Move down**. The precedence order of alias rules is a very important consideration when implementing alias rules.



The scenario above illustrates the importance of precedence. The second alias rule in this scenario (Sql Image/Executable) would never be assigned an alias name if the Sql image resides in the Sql directory.

Process Execution Rules

Process Execution rules offer basic management options such as setting CPU affinity, CPU priority and memory limitations. Process Execution rules are also a requirement for more advanced Process Management, although it is possible to define settings such as Affinity, Priority, Priority Management and Memory Limits in a Process Execution Rule, the recommendation is to manage and control processes and resources with Process Group Execution Rules whenever possible.

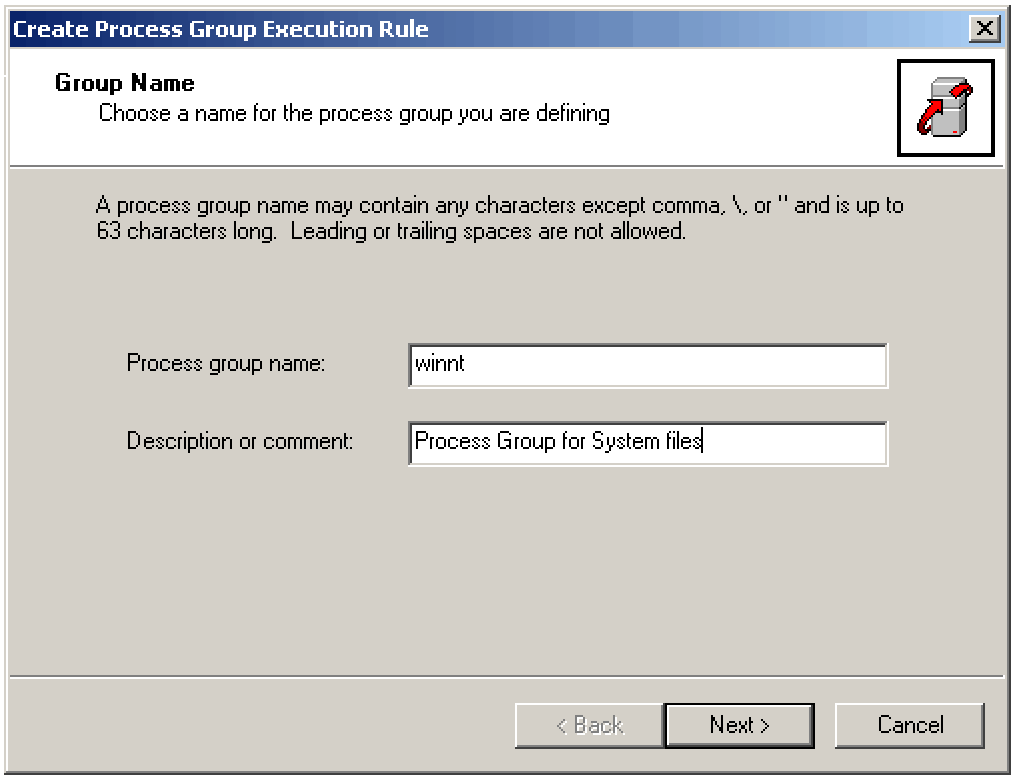
Process Execution Rules allows the usage of Process Group Execution Rules, the core of Process Control.

Process Group Execution Rules

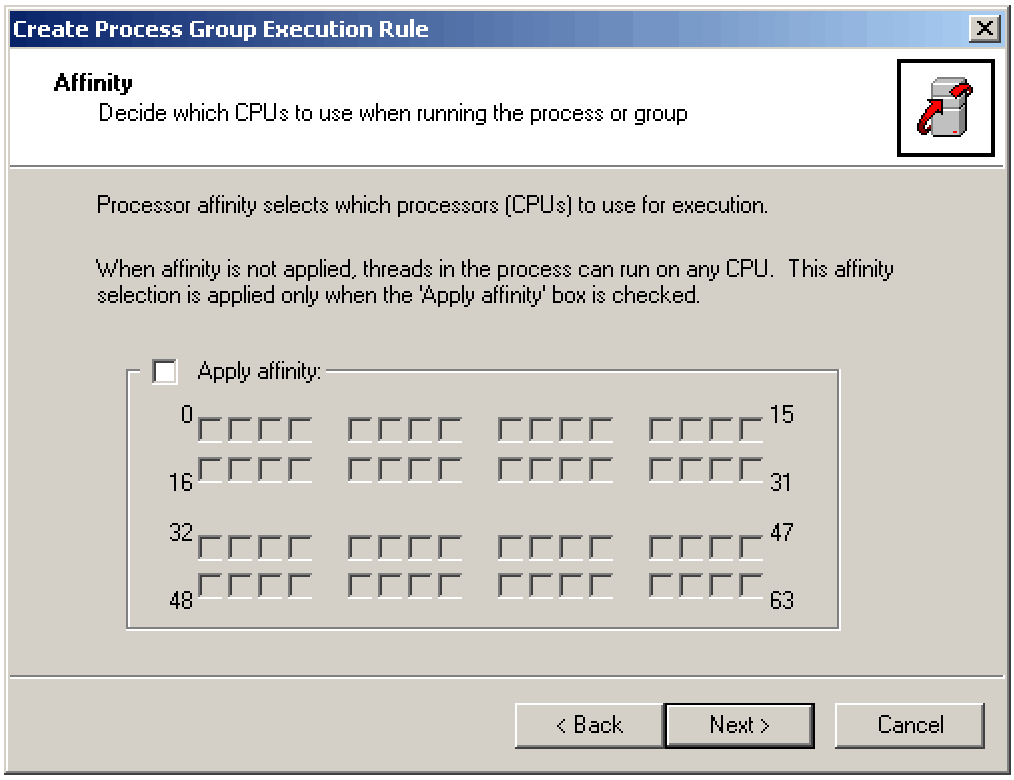
Process Group Execution Rules offer more advanced management options than Process Execution Rules including managing CPU time, memory, and the number of active processes within a group. The key difference between a Process Execution Rules and a Process Group Execution Rules is the usage of Job Objects in Process Group Execution Rules. A Job Object is a Windows 2000 component that allows a process or groups of processes to be managed as one unit. Please see Microsoft's web site for more information about Job Objects.

Process Group Execution Rules require a Process Alias rule associated with a Process Execution Rule, which is set to execute within a Process Group Execution Rule.

A Process Group Execution Rule allows the definition of:

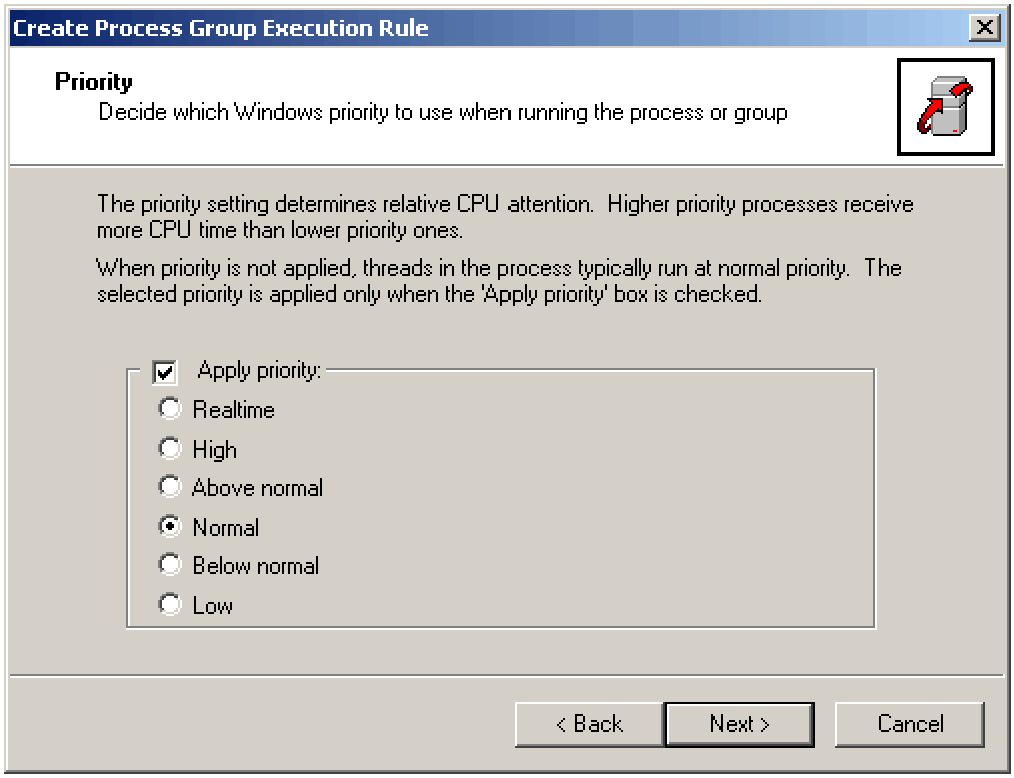


- Process Group Name



- Affinity

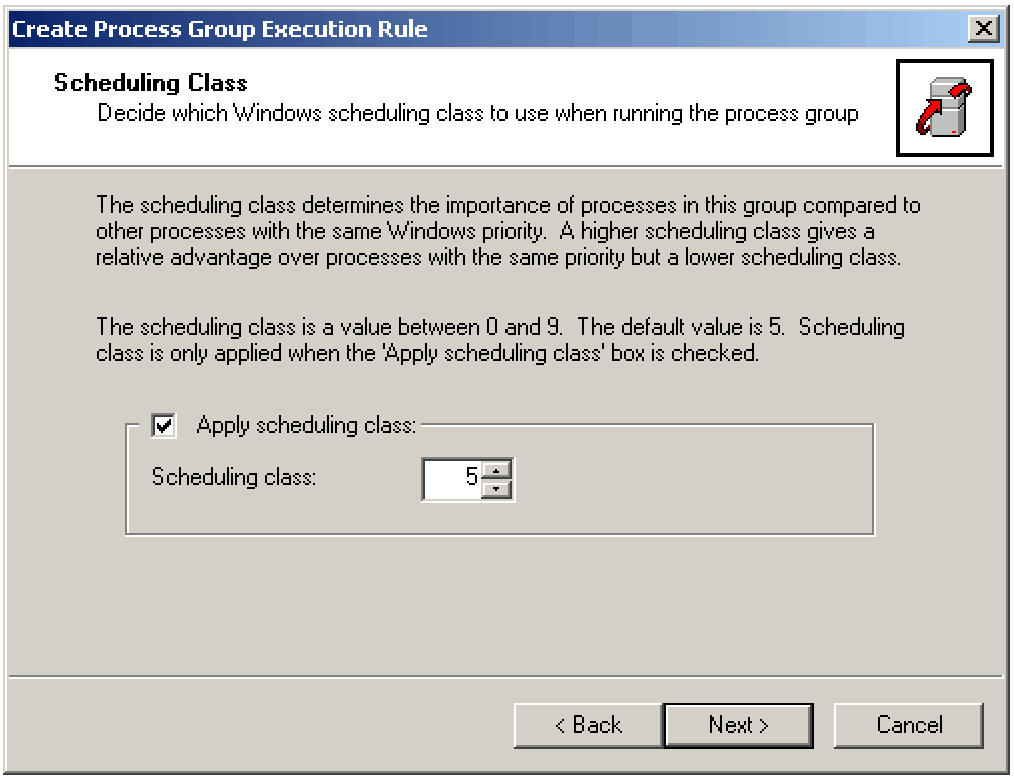
  Use the affinity setting to specify which processor or set of processors a given process can use. For example, a process can be restricted to the first two processors on a four-processor system while another process is restricted to the second two processors. A server can effectively partition available resources in a way that is independent of the application's relative workload by controlling which processors run particular processes or process groups. Moreover, it is possible to change the affinity setting for a given process group to a large share of resources if one process group becomes more active or more important than the others.



- Priority Management

  The Priority setting determines relative CPU attention. Higher priority processes receive more CPU time than lower priority ones. CPU priority can be set to:

  - Real time
  - High
  - Above normal
  - Normal
  - Below normal
  - Low

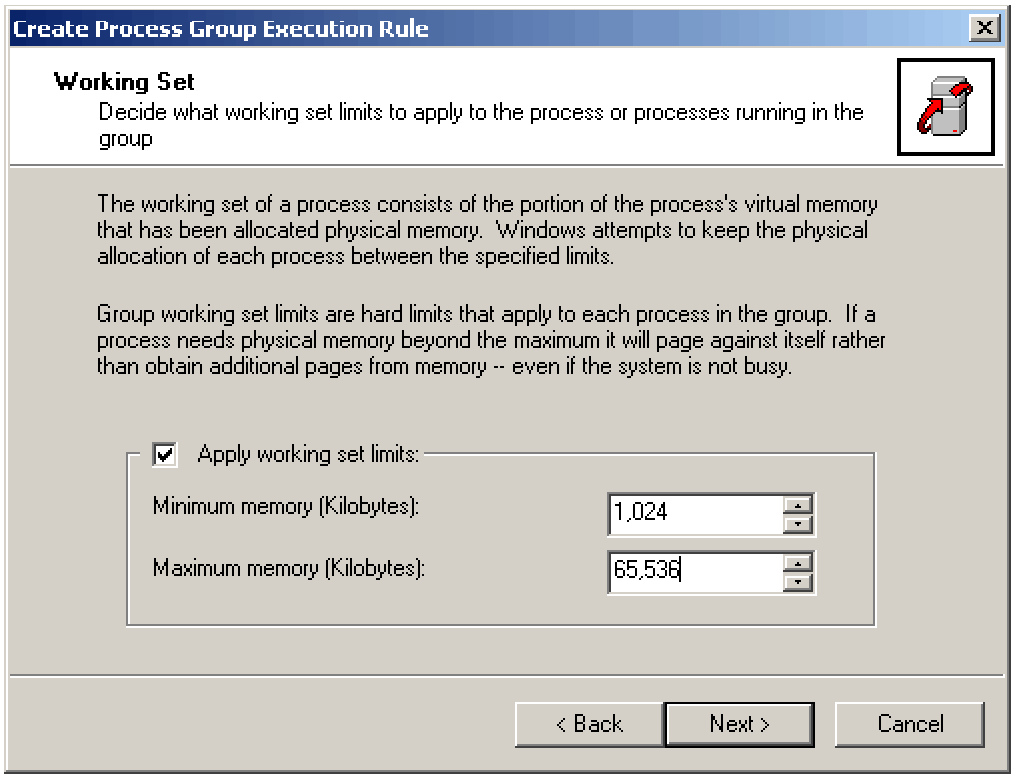


- Scheduling

  The scheduling class determines the importance of processes in this group compared to other processes with the same Windows Priority. A higher scheduling class gives a relative advantage over processes with the same priority but a lower scheduling class. The scheduling class ranges from zero to nine, where zero indicates the least importance and nine the highest.

  Every process running in a process group inherits the scheduling class setting for that particular process group. Processes cannot set their own scheduling class.

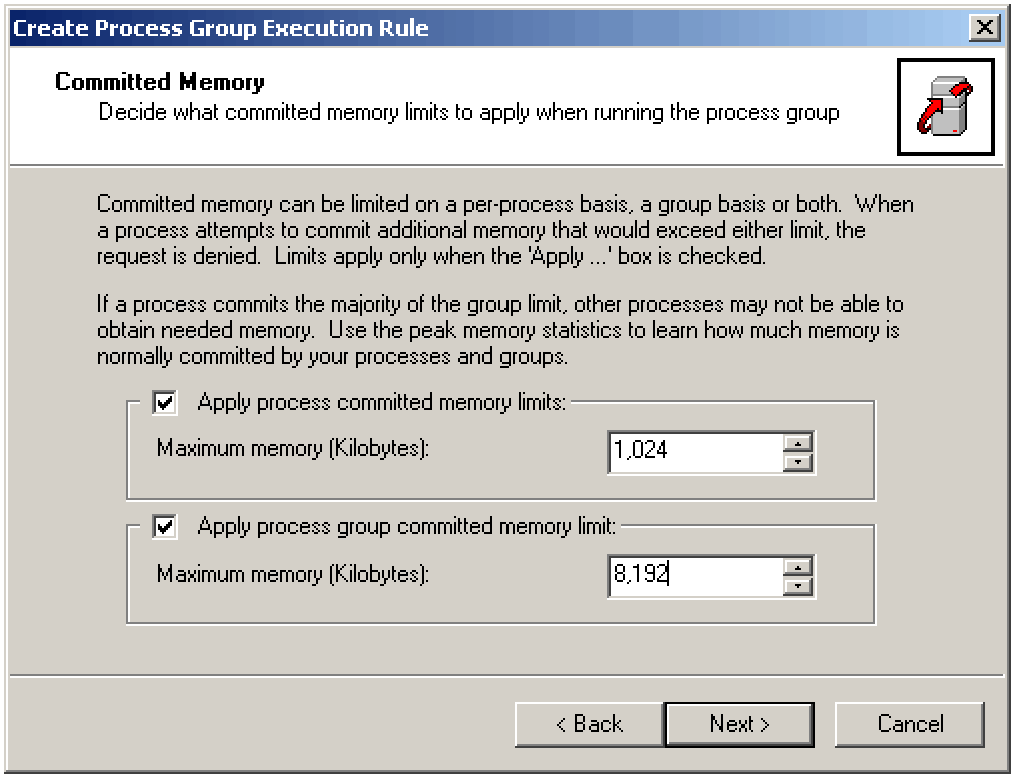
  The default scheduling class value is five. In general, do not use values greater than seven or less than two without deliberate planning and consideration to avoid unexpected results within a given priority group. For example, a scheduling class of zero indicates that the group should not run at all unless the processor has no other work to do in that priority class; a scheduling class of nine indicates that the group should always run (until it becomes idle) when the processor performs work in that priority class.



- Memory

  - Working Set Limits

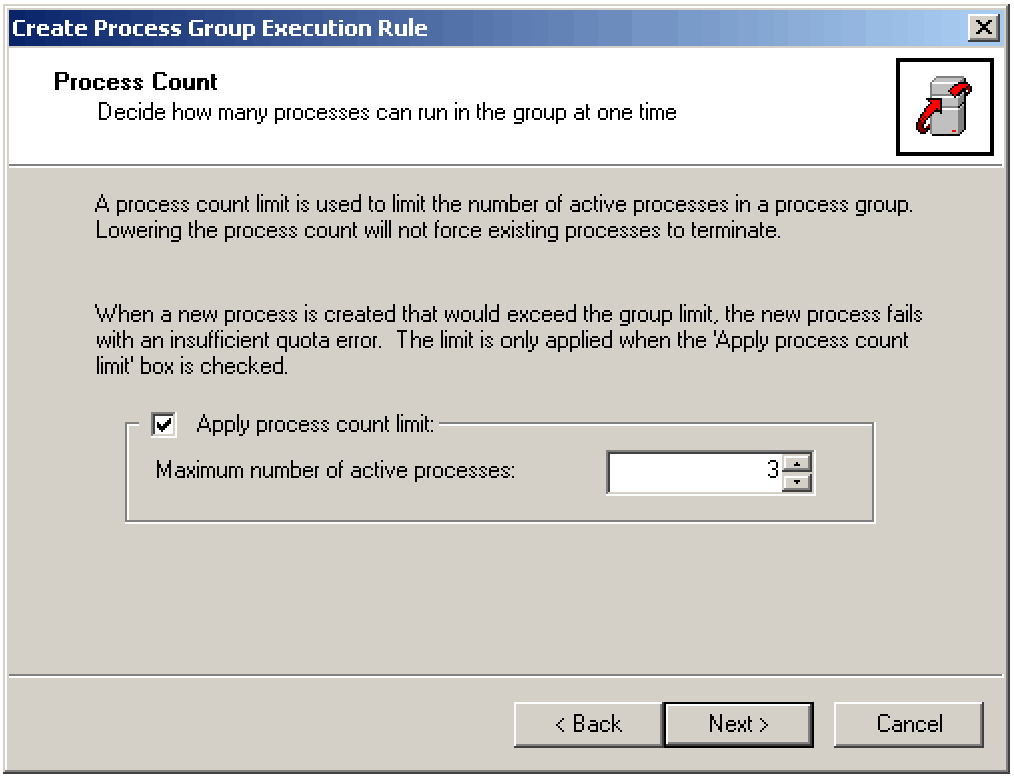
  The working set of a process is the amount of memory physically mapped to its process context. Windows attempts to keep the physical allocation of memory of each process between the specified limits.



- Process and Process Group committed memory limits

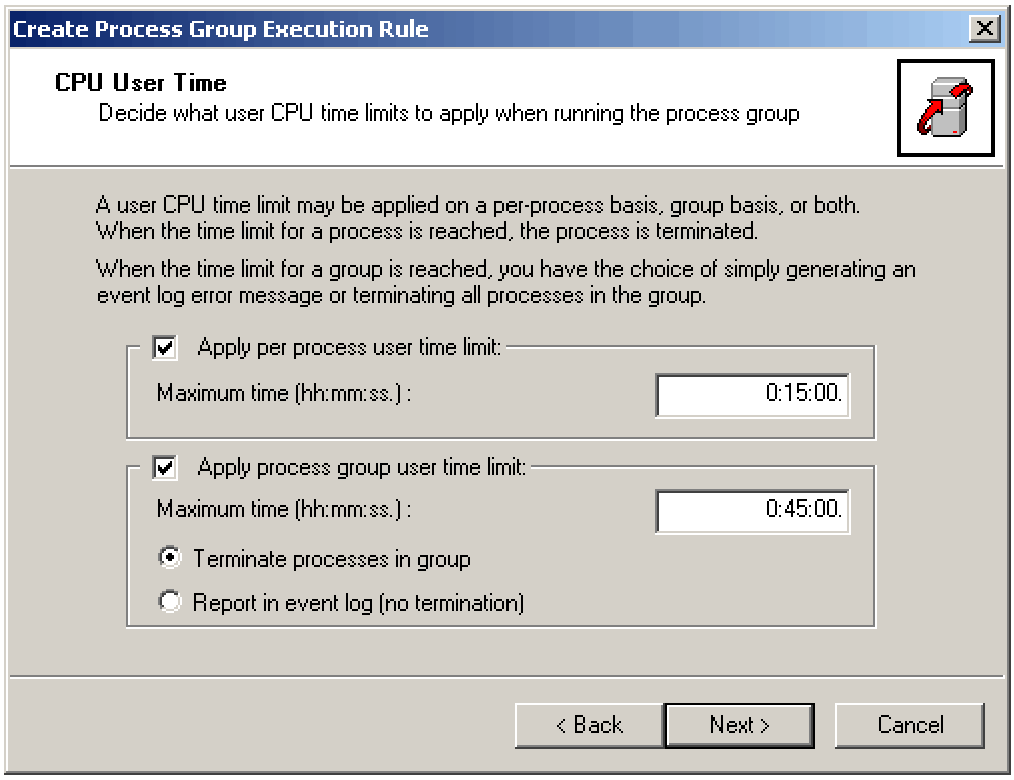
Specifies whether there is a maximum amount of committed memory available to this group at any given time. Committed memory is memory for which the memory manager has corresponding disk storage for a process or a process group

.



- Apply process count limit

  The apply process count limit defines the number of active processes in a group. The interval setting for Process Control define the time a process can execute before the Process Count Limit constraint becomes effective.



- Time

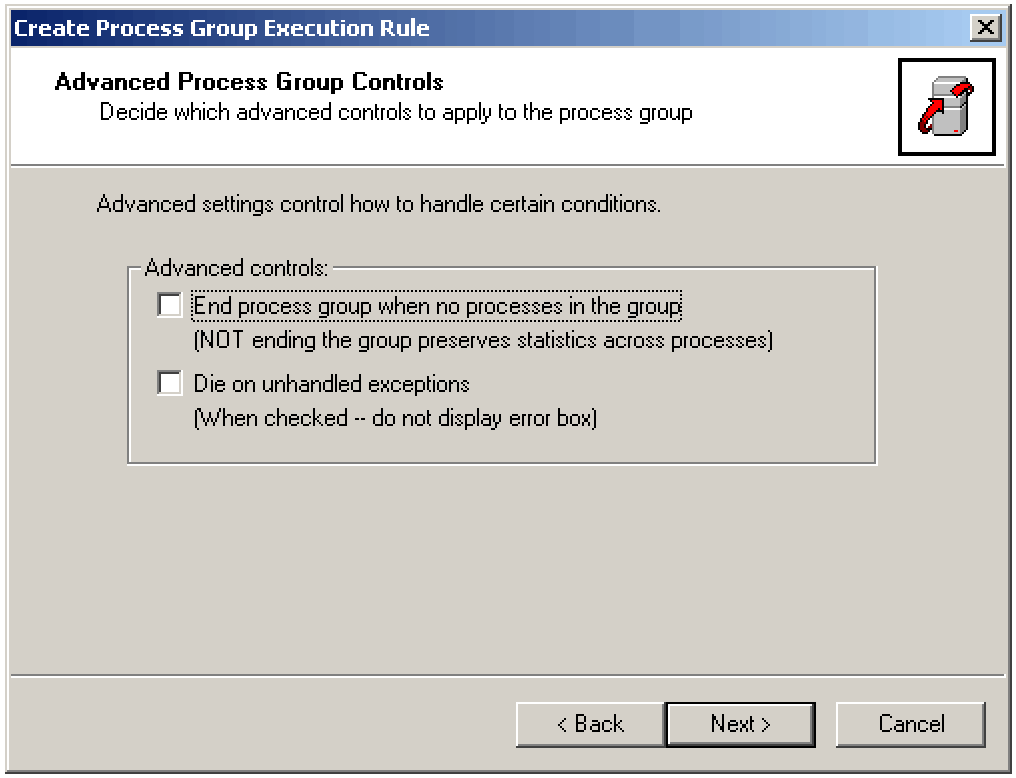
  The ability to set a time limit for a process can limit the execution of runaway processes.

  - Per process user time limit (Maximum time (hh:mm:ss))

  Specifies whether there is a maximum amount of time a given process within this group is allowed to run before it is terminated.

  - Process group user time limit (Maximum time (hh:mm:ss))

  Specifies whether there is a maximum amount of time all processes in this group are allowed to run. If the group exceeds its time limits, the group is terminated and the event is logged, or it is allowed to continue after a system message is logged.



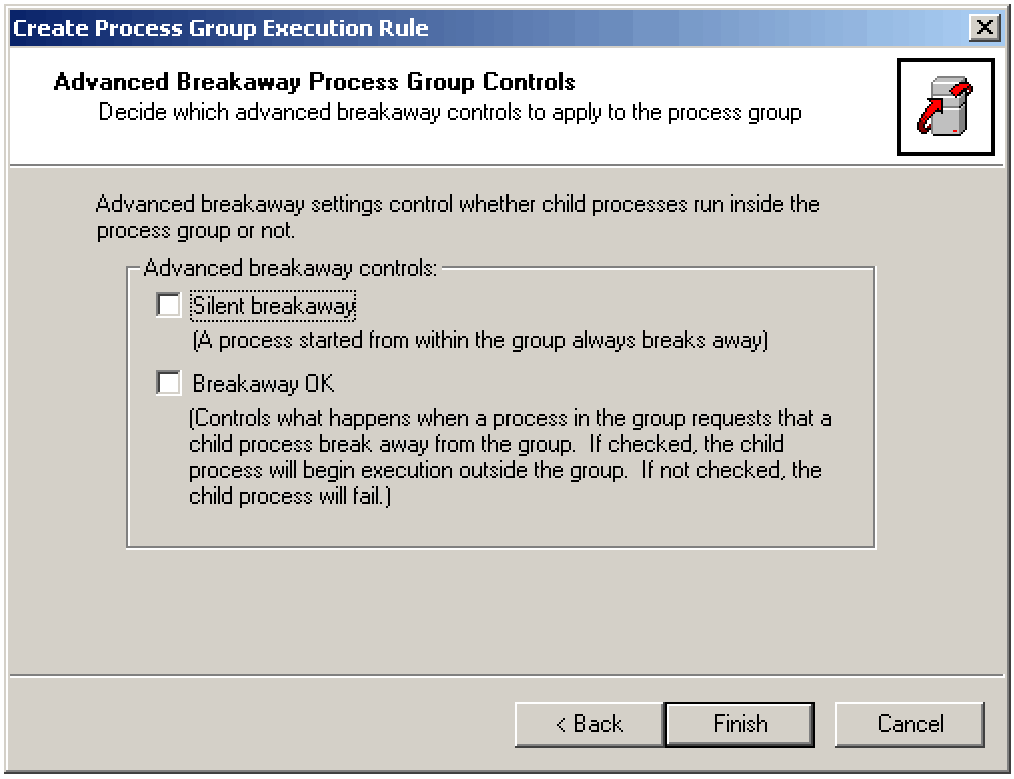
- Advanced

  - End process group when no processes are in the group

  Select this check box to terminate the group when the group is empty. Process Control stops the group from running and deletes any group statistics. Clear this check box to retain group statistics and continue to accumulate data when new processes start and enter the group.

  - Die on unhandled exceptions

  Select this check box to stop warning messages from appearing when an unhandled error exception occurs (appropriate for applications running on unattended servers where there is no operator to see message boxes). Clear this check box to enable the warning message.



- Silent breakaway

Select this check box to permit a process that was originally started by another process in a group to leave the group. Once the process leaves the group, it is eligible to be placed in a group by Process Control during a management scan for example:

A group named Cmd with a Process Alias of Cmd would allow Notepad to leave the Cmd group if the Notepad process was invoked at the Command prompt and the Silent Breakaway checkbox is checked, the Notepad process will remain in the Cmd group if the checkbox remains unchecked.

- Breakaway OK

Select this check box to approve requests by processes in a group to allow processes to break away from the group. Clear this check box to deny those requests and ensure those processes fail. This control works in conjunction with the application requesting the breakaway. If the application resides on a server, and its child processes are not permitted to break away, the application might not work properly. If no application makes such a request, this control option has no effect.

Process Group Execution Rules and Performance Monitor

Process Group Execution Rules are available as counters in Performance Monitor once they are created. The performance object Job Object allows the following counters to be set for a specified Process Group Execution Rule once it's created:

- Current % Kernel Mode Time
- Current % Processor Time
- Current % User Mode Time
- Pages / Sec
- Process Count - Active
- Process Count - Terminated
- Process Count - Total
- This Period mSec - Kernel Mode
- This Period mSec - Processor
- This Period mSec - User Mode
- Total mSec - Kernel Mode
- Total mSec - Processor
- Total mSec - User Mode

The performance object Job Object Details allows the following counters to be set for a specified Process Group Execution Rule once it's created:

- % Privileged Time
- % Processor Time
- % User Time
- Creating Process ID
- Elapsed Time
- Handle Count
- ID Process
- IO Data Bytes / sec
- IO Data Operations / sec
- IO Other Bytes / sec
- IO Other Operations / sec
- IO Read Bytes / sec
- IO Read Operations / sec
- IO Write Bytes / sec
- IO Write Operations / sec
- Page Faults / sec

- Page File Bytes
- Page File Bytes Peak
- Pool Nonpaged Bytes
- Pool Paged Bytes
- Priority Base
- Private Bytes
- Thread Count
- Virtual Bytes
- Virtual Bytes Peak
- Working Set
- Working Set Peak

Please see Performance Monitor in Windows 2000 Server, Advanced Server or Windows 2000 Datacenter for more in depth information regarding Job Object and Job Object Details.

The information derived from a Process Group Execution Rule object in Performance monitor can for example be used to:

- Collect statistics for Charge Back Scenarios, in which a department is charged according to processor and memory utilization
- Automatically alter a Process Control configuration with the built in Command Line Interface once an event is triggered in Performance Monitor for a Process Group Execution Rule
- Interact with a system management solution such as IBM Director

5. Process Control Basic Scenarios and Examples

List active processes on remote system

Purpose: View active processes on remote or local system
Requirements: Standard Process Control installation
Procedure:
1. Open **Process Control.**
To open **Process Control** click **start**, point to **Programs**, point to **Administrative Tools**, and then click **Process Control.**
2. Right-click **Process Control (Local)** and click **Connect to another computer**.
3. Select the **Another Computer** option box in the This Snap-in will manage text box.
4. Type the name of a system or browse to another system.
5. In the console tree, double-click **Process Control** to expand the tree.
6. Click on **Processes**
Results: Process Control displays processes on a remote computer

View processes by directory

Purpose: Determine where a certain process started, which can be vital information when troubleshooting a System
Requirements: Standard Process Control installation
Procedure:
1. Open **Process Control.**
To open **Process Control** click **start**, point to **Programs**, point to **Administrative Tools**, and then click **Process Control.**
2. In the console tree, double-click **Process Control** to expand the tree.
3. Double-click **Rules.**
4. Click **Process Alias Rules.**
5. In the details pane, right-click the default rule, then click **Insert Rule.**
6. In the **Type of match** option box select **Sub-Directory Name**, in the **Match string** text box, type ***.
7. In the **Process Alias assigned on a match** text box, type **<D>**, or click on the **arrow** button and select **Substitute Sub-Directory Name.**
8. Click **OK.**
Results: Process Control displays processes based on the directory in which a process originated or started

Differentiate processes with the same image name

Purpose: Differentiate a production application and test application running on the same system with the same image name but executed from different directories.
Requirements: Standard Process Control installation
Procedure:
1. Open **Process Control.**
To open **Process Control** click **start**, point to **Programs**, point to **Administrative Tools**, and then click **Process Control.**
2. In the console tree, double-click **Process Control** to expand the tree.
3. Double-click **Rules.**
4. Click **Process Alias Rules.**
5. In the details pane, right-click the list to insert the new rule, then click **Insert Rule.**
6. Define the new Process Alias Rule.
In the **Type match,** select the **String** check box.
In the **Match string** text box, type: <directory\process>
In the **Process alias assigned on a match,** type: <alias string >
7. Click **OK.**
8. Repeat steps 4-7 for each additional process with the same image name
Results: Two or more processes executed from different directories can be differentiated from each other by their alias name in the Process list in Process Control.

Hide processes from view

Purpose: Set emphasis on important processes by filtering less important processes.
Requirements: Standard Process Control installation
Procedure:
1. Open **Process Control.**
To open **Process Control** click **start**, point to **Programs**, point to **Administrative Tools**, and then click **Process Control.**
2. Click **Processes**. On the **View** menu, click **Show Only Running Processes** or **Show Only Managed Processes.**
Alternative:
1. Open **Process Control.**
To open **Process Control** click **start**, point to **Programs**, point to **Administrative Tools**, and then click **Process Control.**
2. In the console tree, double-click **Process Control** to expand the tree.
3. Double-click **Rules.**
4. Click **Process Alias Rules.**
5. Right-click selected **Rule**, then click **Edit**.
6. In the **Process Alias assigned on a match** text box, add <H> or click on the **arrow button** and click on **Hide**.
7. Click on **OK.**
Results: Only filtered processes can be viewed in the process list

Limit the number of active processes in a group

Purpose: Enforce control of active processes within a group, regardless of the actual number of processes needed for a certain application.
Requirements: Standard Process Control installation
Procedure:
1. Open **Process Control**
To open **Process Control** click **start**, point to **Programs**, point to **Administrative Tools**, and then click **Process Control.**
2. In the console tree, double-click **Process Control** to expand the tree
3. In the **Rules** tree, click on **Process Groups**
4. In the details pane, right-click the process group to edit. Then click **properties**.
5. **Select** the **Apply process count limit** text box and add the number of active processes for the group.
Results: The number of active processes in this group has been restricted

Display accumulated group statistics

Purpose: Display accumulated group statistics for charge back or usage statistics.
Requirements: Standard Process Control installation
Procedure:
1. Open **Process Control**
To open **Process Control** click **start**, point to **Programs**, point to **Administrative Tools**, and then click **Process Control.**
2. In the console tree, double-click **Process Control** to expand the tree
3. In the **Rules** tree, click on **Process Groups**
4. In the details pane, right-click white space, then click **export list**.
5. When prompted to save statistics to a text file, select/type file name and destination
6. Click on **Save**
Results: A tab delimited text file has been created containing data and statistics regarding Process Groups and Processes. Statistics recorded in this file includes categories such as Status, Active Processes, Affinity, Priority, Scheduling Class, User Time, Kernel Time, Page Faults, Process Count, Terminated Processes, Read Operations, Write Operations, Other Operations, Read Transfer Bytes, Write Transfer Bytes, Other Transfer Bytes, Peak Process Memory Used, and Peak Group Memory Used.

Set working set limits for process groups

Purpose: Prevent applications with memory leaks from allocating new memory every time it executes.
Requirements: Standard Process Control installation
Procedure:
1. Open **Process Control.**
To open **Process Control** click **start**, point to **Programs**, point to **Administrative Tools**, and then click **Process Control.**
2. In the console tree, double-click **Process Control** to expand the tree.
3. Double-click **Process Groups.**
4. In the Process Groups tree, right click **"selected group"** and click **properties.**
5. Click the **Memory** tab.
6. Define the new Working Set Limits.
Select the **Apply working set limits** check box.
In the **Minimum memory** text box, type: <Minimum memory in kilobytes>
In the **Maximum memory** text box, type: <Maximum memory in kilobytes>
Click **OK.**
Results: Memory constraints prevent applications from allocating memory beyond selected memory size.

Set physical memory limits for a group of processes

Purpose: Specify how much memory a process can use.
Requirements: Standard Process Control installation
Procedure:
1. Open **Process Control.**
To open **Process Control** click **start**, point to **Programs**, point to **Administrative Tools**, and then click **Process Control.**
2. In the console tree, double-click **Process Control** to expand the tree.
3. Double-click **Process Groups.**
4. In the Process Groups tree, right click **"selected group"** and click **properties.**
5. Click the **Memory** tab.
6. Define the new Physical memory Limits.
Select the **Apply process committed memory limits** check box.
In the **Maximum memory** text box, type: <Maximum memory in kilobytes>
Click **OK.**
Results: A memory constraint has been defined preventing a process from using more than specified committed memory at any given time.

Set CPU user time limits for a process

Purpose: Limit the execution time for a specific process or group of processes.
Requirements: Standard Process Control installation
Procedure:
1. Open **Process Control.**
To open **Process Control** click **start**, point to **Programs**, point to **Administrative Tools**, and then click **Process Control.**
2. In the console tree, double-click **Process Control** to expand the tree.
3. Double-click **Process Groups.**
4. In the **Rules** tree, click **Process Group Execution Rules.**
5. In the details pane, double-click **<name of group>**
6. Change the user time limit for the group.
Click the **Time** tab.
Select the **Apply process group user time** box.
In the Maximum time text box, type: <Maximum Time in (hh:mm:ss)>
Select the **Terminate processes in group** check box.
Click **OK.**
Results: Processes in this group will now terminate after specified CPU user time have elapsed.

Manage affinity for a group of processes

Purpose: Control under which CPU's a number of processes should run.
Requirements: Standard Process Control installation
Procedure:
1. Open **Process Control.**
To open **Process Control** click **start**, point to **Programs**, point to **Administrative Tools**, and then click **Process Control.**
2. In the console tree, double-click **Process Control** to expand the tree.
3. Double-click **Rules.**
4. Click **Process Alias Rules.**
5. In the details pane, right-click the list to insert the new rule, then click **Insert Rule.**
6. Define the new Process Alias Rule.
In the **Type match,** select the **Image Name** check box.
In the **Match string** text box, type: <string to match>
In the **Process alias assigned on a match,** type: <alias string>
7. In the **Rules** tree, right-click **Process Execution Rules** and click **New Process Execution Rule.**
8. Define the new Process Execution Rule.
In the **Process alias** text box, type: <previously entered alias string>
Click **Next.**
Select the **Execute within a process group** check box.
In the text box, type: <name of group>
Click **Next.**
Click **Next.**
Click **Next.**
Click **Finish.**

9. In the **Rules** tree, click **Process Group Execution Rules.**
10. In the details pane, double-click **<name of group>**
11. Change the affinity for the group.
Click the **Affinity** tab.
Select the **Apply affinity** box.
Check the first box in the group.
Click **OK.**
Results:
All processes in this group will now run on the first CPU.

6. Process Control Advanced Scenarios and Examples

Consolidate Server Applications

Purpose: Minimize the cost for hardware while providing secure and reliable Server application availability. Process Control provides the tool to achieve Server Consolidation it is however, necessary to have a baseline and thorough knowledge of resource usage for applications you are planning to consolidate before applying constraints for processes in Process Control.
Requirements: Standard Process Control installation, Baseline of an applications memory and cpu usage over a period of time.
Procedure: See the Usage Guide section for setting CPU affinity and memory constraints.
Results: Lower TCO (Total Cost of Ownership)

Prevent unauthorized applications from running

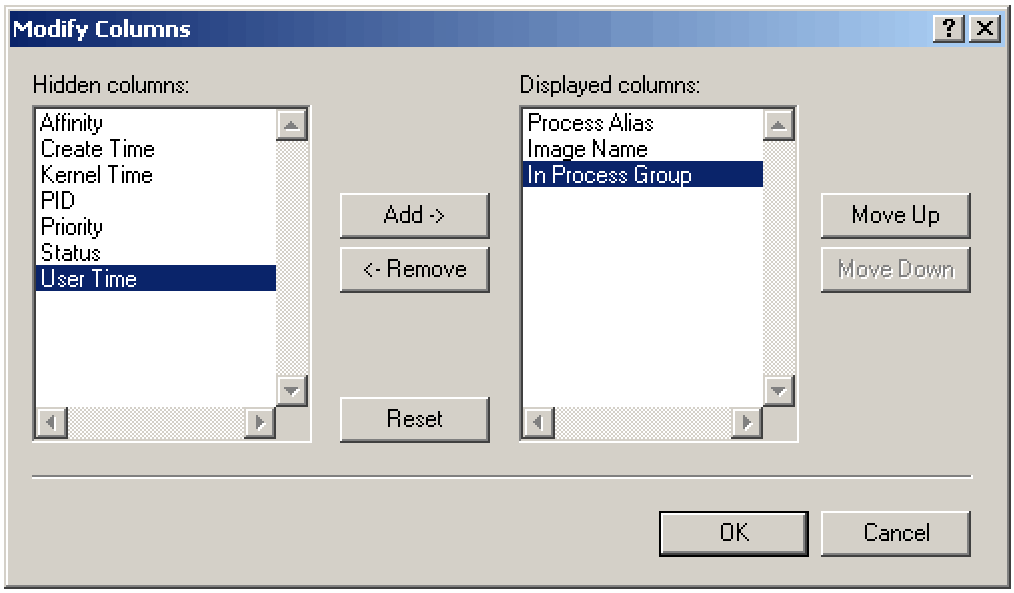
Purpose: Secure your servers from unauthorized application usage.
Organizations can secure their servers from virus threats, unauthorized applications and malicious code by reversing the default rule and restricting processes to a limited number of directory matches.
Please note that this usage model can potentially cause a server to stop responding or malfunction if applied incorrectly.
Requirements: Standard Process Control installation
Use the View Processes by Directory guide to list process – directory dependencies. These dependencies are necessary when creating rules for directory match.
Proceed with the guide below when authorized directories and processes have been identified
Procedure:
1. Open **Process Control**
To open **Process Control** click **start**, point to **Programs**, point to **Administrative Tools**, and then click **Process Control.**
2. In the console tree, double-click **Process Control** to expand the tree
3. Double-click **Rules.**
4. Click **Process Alias Rules**
5. In the details pane, right-click the list to insert the new rule, then click **Insert Rule.**
6. Define the new Process Alias Rule.
In the **Type of Match** option box, select **Sub-Directory Name**
In the **Match string** text box, type for example: system32
In the **Process alias assigned on a match** text box, type for example: system32
Click **OK.**
7. Create additional alias rules based on **Sub-Directory Name match**.
8. In the details pane, right-click the previously created Process Alias Rule, then click **Move Up.**
9. Create the last alias rule based on **Sub-Directory Name match**.
In the **Type of Match** option box, select **Sub-Directory Name.**
In the **Match string** text box, type “*”
Click **OK.**

A number of alias rules have been created based on Directory Matches. It is very important to move authorized directories and processes to the top of the list and the “*” last.
10. In the **Rules** tree, right-click **Process Execution Rules** and click **New Process Execution Rule.**
11. Define the new Process Execution Rule.
In the **Process Alias** text box, type: “*”
In the **Description or comment** text box, type: Unauthorized
Click **Next**
Select the **Execute within a process group** check box.
In the **Create a process group** text box type: Unauthorized
Click **Next**
Click **Next**
Click **Next**
Click **Finish**
11. In the **Rules** tree, click on **Process Groups**
12. In the details pane, right-click the process group named “*”, then click **properties**.
13. **Select** the **Apply process count limit** text box. The **Maximum number of active processes** should be equal to 0.
Use caution when implementing the above solution; it will not allow any processes to run outside the previously defined Process Alias Rules directories.
Results: Selected applications and processes have been given the right to execute while remaining applications and processes have been denied.

7. Appendix A: How-To:

Set emphasis on selected columns in Process Control

Purpose: Emphasize selected columns for display in Process Control
Requirements: Standard Process Control installation
Procedure:
1. Open **Process Control**
To open **Process Control** click **start**, point to **Programs**, point to **Administrative Tools**, and then click **Process Control.**
2. In the console tree, double-click **Process Control** to expand the tree
3. Double-click **Processes.**
4. Click **View** on the menu bar, then **Choose Columns**
5. In the **Modify Columns** dialogue box select/deselect which columns you wish to display.
6. Click **OK.**
Results: Only Process Alias, Image Name and In Process Group columns will be displayed under the Process



Backup a Process Control Rule Configuration

Purpose: Minimize Process Control Rule configuration when Re-Installing OS image
Requirements: Standard Process Control installation
Procedure:
1. Open a **command prompt**
2. In the **command prompt** type **cd** and then press **enter**
3. In the **command prompt** type **cd program files\ibm\process control** and then press **enter**
4. In the **command prompt** type **proccon –xd d:\backups\proccon.txt**
Results: Process Alias Rules, Process Execution Rules and Process Group Execution Rules have been backed up successfully.

Restore a Process Control Rule Configuration

Purpose: Restore a Process Control Rule Configuration after an OS crash
Requirements: Standard Process Control installation
Procedure:
1. Open a **command prompt**
2. In the **command prompt** type **cd** and then press **enter**
3. In the **command prompt** type **cd program files\ibm\process control** and then press **enter**
4. In the **command prompt** type **proccon –xr d:\backups\proccon.txt**
Results: Process Alias Rules, Process Execution Rules and Process Group Execution Rules have been restored successfully.

Install Process Control

Process Control is distributed as an Installshield package on IBM's web.
Double-Clicking **ProcConPackage.Exe** will invoke an installshield wizard through the remainder of the installation procedure.
The following installation types can be selected during the installation wizard:

- Custom Installation
  The following Process Control components can be selected during a Process Control installation:
  - Server
    The Server component installs the Service and Mediator, components required for any computer that is being managed
  - Clients
    The Clients component installs the Microsoft Management Console Snap-in (MMC), the Command Line Utility or both. The Clients components are required to manage any Process Control system with the Server components installed.
- Typical Installation
  Process Control will install all components

A typical Process Control installation consists of the following files:

- Proccon.dll
  Proccon.dll is a Microsoft Management Console (MMC) Snap-In working as a client to the Process Control Service.
- Proccon.exe
  Proccon.exe is the Command Line Interface for Process Control.
- Proccon.msc
  Proccon.msc is a pre-configured Process Control Microsoft Management Console (.msc) file.
- Procconmd8.exe
  Procconmd8.exe is controlled by the Process Control service, it contains and preserves information when the Process Control service starts and stops.
- Procconsvc.exe
  Process Control Service, must be installed on all target systems
- Readme.txt
  Readme.txt contains last minute information about the latest Process Control version.
- Proccon.chm
  Proccon.chm is a HTML Help file and an excellent resource for further information about Process Control.

* Note that both Procconmd8.exe and Procconsvc.exe are referred to as Services in some documentation but only Procconsvc.exe answers to Net Stop and Net Start commands.

Uninstall Process Control

Process Control can be uninstalled by double-clicking the **ProcConPackage.Exe** or by clicking on **Process Control** in the list of **Change or Remove Programs** in the **Add/Remove Programs** applet in the **Control Panel**.

Modifying or repairing an existing Process Control installation

Double-Clicking **ProcConPackage.Exe** will invoke the Process Control Maintenance program. This program allows the modification or reparation of an existing Process Control installation.

8. Appendix B: Troubleshooting

This section provides the most frequently asked questions & problem-solving guidelines for Process Control. The section down below can also be found in the online help in Process Control.

FAQ regarding Process Control

1. Processes fail to show up in a group

Cause: You created an Alias Process Rule with a different name than that defined in Process Execution Rules.

Solution: Create an Alias Process Execution Rule with the same Alias name as defined in Process Alias Rule dialog.

2. MMC reports "Snap-In failed to initialize. Name: Process Control".

Cause: MMC not finding the Process Control Snap-In "ProcCon.dll" is the most likely cause.

Solution: If the Process Control Snap-In "ProcCon.dll" has been moved from its original location it needs to be reinstalled (re-registered) or moved back to its original location. ProcCon.dll is usually installed in "Program Files\Ibm\Process Control\".

3. Process Control does not show up in Computer Management.

Cause: The Process Control service is not installed on the computer being managed.

Solution: Verify that the Process Control service is installed on the computer being managed. Process Control is a dynamic extension of Computer Management. If the Process Control service is not installed on the computer being managed, Process Control will not appear in the Services and Application folder in the console tree.

4. Process Control does not give the option to connect to another computer.

Cause: You are running the Process Control snap-in from the Computer Management console.

Solution: When you run Process Control from the Computer Management console, Process Control and Computer Management will manage the same computer. To manage another computer, in the console tree, click Computer Management, and then from the Action menu, click Connect to another computer.

Alternate Solution: Use the stand-alone version of Process Control. Choose Process Control from the Start menu, where your other administrative tools, such as Computer Management, are located. This way, you can direct Process Control at any computer, regardless of the Computer Management target.

5. The Service tab for Process Control properties does not always appear.

Cause: The Process Control snap-in cannot communicate with the Process Control service.

Solution: Make sure the Process Control service is running on the computer you are managing.

6. Changes or requests are reported to time out, but the change occurs.

Cause: The Process Control snap-in quit waiting for a response from the Process Control service. Although the Process Control service got the request and acted on it, the response to the snap-in took too long, and the snap-in stopped waiting for a reply.

Solution: If the problem occurs frequently, increase the request time-out interval. The message might be an indicator that the computer being managed is very busy. Note that no harm results from this. Displayed information may be out-of-date (use "refresh" to check this), but you can continue to use the snap-in after a time-out error.

Note: When a time-out error occurs, you cannot tell whether the server received the request, acted on it, or rejected it. Thus, you should look at the data on a refreshed view before deciding whether you should resubmit your request.

7. Changes or updates fail because a database update occurred between the time you retrieved data and the time you attempted to update it.

Cause: The Process Control service has detected and prevented an attempt to update a rule that has just been updated.

Solution: This is not necessarily a problem. Process Control is careful not to allow two or more updates to a single rule at the same time. If multiple system administrators are updating Process Control rules, it may be necessary to coordinate the updates. You can also see this warning if you use multiple Process Control sessions or views from MMC to manage the same computer.

8. Requests take a long time or do not work.

Cause: The Process Control snap-in timed out trying to communicate with the Process Control service on the computer to be managed. The Process Control service did not get the request because the Process Control snap-in gave up or quit trying to communicate.

Solution: If the problem occurs frequently, increase the request time-out interval. The message may be an indicator that the computer being managed is very busy.

Note: When a time-out occurs, you cannot tell whether the server received, acted on, or rejected the request. Thus, you should look at the data on a refreshed view before deciding whether your request needs to be resubmitted.

9. Processes are not being put in process groups, or are put in the wrong process groups.

Cause: If the process was started by another process (its parent), and the parent process is in a process group, the new process will typically be in the same group as its parent process. For example, if the command prompt process Cmd.exe is in a process group, any process, program, or application started at a command line will typically remain in the same process group as its parent process.

Solution: If you want processes to remain in the group where the parent is running, you need not include rules for the new process. Automatically running in the group of the parent process is normal process group behavior. If you want a process that starts other processes (such as Cmd.exe or Explorer.exe) to be included in a group, but either you do not want to include their launched processes in the group or you want them to be eligible for inclusion in other groups, use the silent breakaway feature. Once a process is placed in a process group, the process cannot leave the process group. Thus, it is

important to ensure that any process you want to assign to a group is created free of any group association.

10. Process Control causes applications to run incorrectly.

Cause: When Process Control manages processes and applies certain rules, some applications will behave differently. In some cases, applications will not work within those rules or applied constraints. Memory limits for process groups are stringent; applications are not allowed to break those limits.

Solution 1: Memory Limit is in effect. An application can fail if Process Control limits the committed memory to a value below that which the application requires. You can solve this by turning off (or possibly increasing) the memory limit.

Solution 2: Process count limit is in effect. If the application that does not work properly creates other processes, and the group has a process count limit, other processes might be failing because the limit was exceeded. The processes that continue running might not work properly because other processes failed.

Solution 3: Breakaway OK is required. The process in question might be creating other processes and requesting that they break away from the group. If the group does not permit this because the Breakaway OK setting is off, the new process will fail, and the application will not run correctly.

Solution 4: A group-wide limit is in effect. Group memory and time limits are possible. Such limits are not applied to each process, but are applied to the group as a whole. Thus, if one process in a group acquires most of the allowed memory, another process might fail to acquire even the basic amount of memory needed to begin running. If a group's time limit is exceeded, a process that has used very little time may be terminated along with the process that used most of the time. Do not use group-wide limits unless it makes sense to do so for the mix of processes in the group.

Solution 5: Working set limit is mismatch. When a working set limit is in effect for a group, that limit is applied to each process in the group. A limit that is ample for one process could be inadequate for another process. Because this is a hard limit, the process that requires more memory will page against itself and appear to be performing poorly, even though the system as a whole and other processes in the group are performing well. Do not use group working set limits for a mix of processes that require widely different working sets.

11. Process Control causes other applications to fail or receive access violations.

Cause: Memory limits for process groups are stringent, and applications are not allowed to break those limits. Thus, if an application asks for memory and a process group has a memory limit in place, the application's request for memory will fail if the request would push memory past the limit. An application can get an access violation as a result of using nonexistent memory.

Solution: Each application should check that memory allocation requests are successful. If the application in question does not perform such checks, do not apply memory limits, or the application might fail when the limit is reached.

12. Process Control fills the event log.

Cause: Many updates are being made to Process Control data, managed groups are hitting many limits, or many processes are being added to groups (or are being started from within groups).

Solution: Examine the messages to determine whether a group is hitting memory limits frequently or has many processes starting and stopping in the group. Adjusting or removing memory limitations might solve the problem. The system log size can be expanded, and the log can be set to wrap when full.

13. Process Control causes applications to run more slowly.

Cause: Process Control is applying rules that lower the priority, limit the number of processors to be used, or restrict the amount of memory the application is using.

Solution: Modify or remove Process Control rules that are slowing the process. Note, however, that in a busy system, rules that permit a particular process or group to consume more resources can slow other processes or groups.

14. Process Control does not allow processes to run.

Cause: Process Control is applying a rule that may be too restrictive.

Solution: In a busy system, processes or groups that belong to a lower priority or scheduling class can be deprived of memory. If there is not enough time for a lower-priority process to run, it will either have to compete on an equal footing by having its priority raised (or other priorities lowered), or processors will have to be apportioned so the lower-priority work gets some attention. For example, with eight processors, one processor might have to be set-aside for low-priority groups, while the higher-priority groups will then compete to use the other seven processors. This way, the low-priority work will get some time and cannot be completely starved by the other work. If the starved process or group is run infrequently (or spends considerable time waiting), it might be desirable to place it at a higher priority than other work. That way, its work will be carried out sooner, other work will be minimally affected, and overall throughput will increase.

15. Task Manager and Process Control do not show the same list of processes.

Cause: The Process Control snap-in can be set to show only managed processes.

Solution: Set the snap-in to show all processes or all running processes. To do this, in the console tree, under Process Control, click Processes. On the View menu, click Show All Processes.

Alternate Cause: A Process Alias rule might be hiding processes.

Solution: Change or remove any Process Alias rules that are hiding processes.

Alternate Cause: Process Control uses standard APIs and might be constrained by security restrictions that an application has placed on its processes, or by other security policies put in place by the system administrator. Task Manager is part of the operating system and uses methods internal to Windows that are not restricted by the same constraints.

Solution: None.

16. Task Manager and Process Control do not show the same priority.

Cause: Windows uses 32 priority levels internally. Externally there are six levels. Thus, when a process is displayed with one of the six priority classes, the 32 possible values are mapped to six. In some cases, the mapping used by Process Control and the mapping used by Task Manager might not be identical.

Solution: None. This is not a problem. Any process in which the class appears with different values actually has a priority somewhere between the two values shown.

17. Processes remain in a group even after the rule is changed.

Cause: Once a process is placed in a process group, the process cannot leave the process group. This behavior is mandated by the underlying Job Object technology in Windows 2000.

Solution: If the group change must be performed immediately, the only solution is to end the process in question and then restart it. Otherwise, the next time the process starts; it will obey the rules as updated.

18. Processes do not exhibit the behavior specified in the Process Execution rule.

Cause: A process can override a Process Execution rule. When the Process Control service recognizes a new process with a Process Execution rule, the service will apply the rule. The Process Control service will continuously try to keep enforcing the Process Execution rule, but between the time a process changes its own setting and the time the Process Control service checks these settings, the limit might not be set as desired.

Solution: If you do not want the process to be able to override a Process Execution rule, put the process in a process group by creating a Process Group Execution rule.

19. Process Control rules do not appear to be taking effect.

Cause: The Process Control service has not yet checked for new processes to manage.

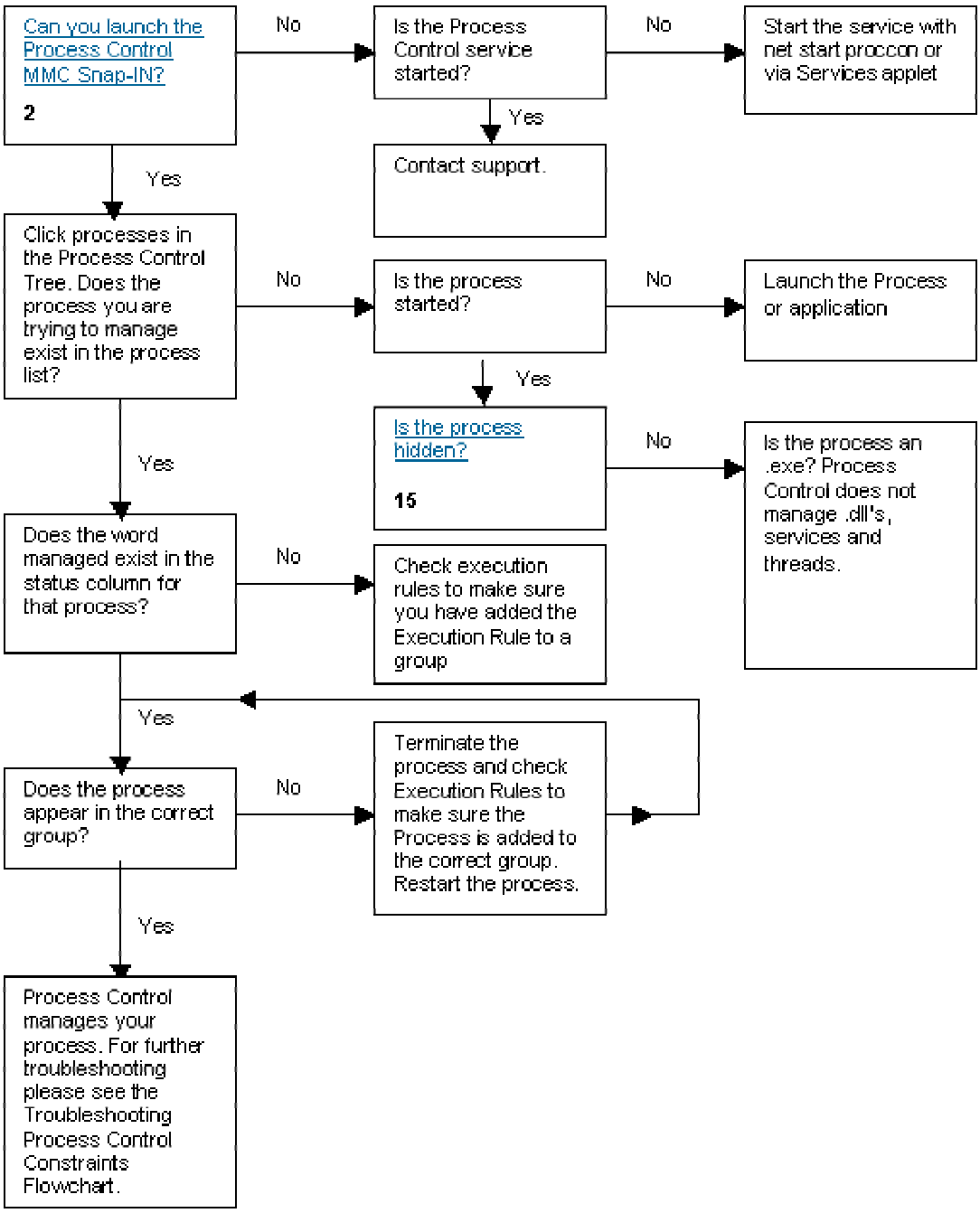
Solution: Wait for the Process Control scan interval to expire, or set a shorter scan interval. Process Control only checks for new processes periodically. This interval is controlled by the scan rate set in the Process Control service. A shorter interval will cause Process Control to check for new processes more often at the expense of slightly increased Process Control overhead.

Alternate Cause: The computer being managed is very busy and the Process Control service is not getting enough time to perform its task.

Solution: It is recommended that the Process Control service be run at high- or above-normal priority. Process Control is not CPU-intensive, but giving it a higher priority ensures that on a busy server, Process Control will be given time to apply any newly defined rules.

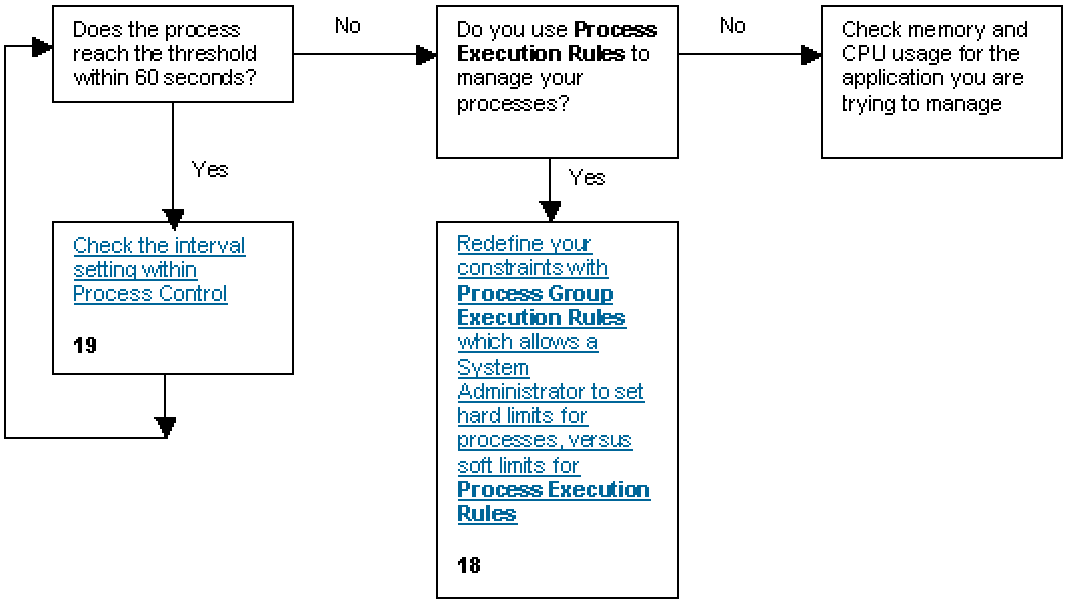
Troubleshooting Process Control Flowcharts

Troubleshooting Process Control Rules



Troubleshooting Process Control Constraints

Note: The number designations in the flowcharts below correspond to the Troubleshooting FAQ.



9. Legal

Produced in the USA
12-15

IBM reserves the right to change specifications and other product information without prior notice. This publication could include technical inaccuracies or typographical errors. IBM makes no representations or warranties regarding third-party products or services. References herein to IBM products and services do not imply that IBM intends to make them available other countries. IBM PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

IBM @server systems are assembled in the U.S., Great Britain, Japan, Australia and Brazil and comprise U.S. and not U.S. components.

IBM, the IBM logo, the e-business logo and xSeries are registered trademarks or trademarks of International Business Machines Corporation in the United States and/or other countries.

Intel is a registered trademark of Intel Corporation.

Microsoft, Windows, Windows NT and the Windows logo are trademarks or registered trademarks of Microsoft Corporation.

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