

Tech Note 719

Optimizing Wonderware® Application Server Performance on Windows® 7/Server 2008 R2

All Tech Notes, Tech Alerts and KBCD documents and software are provided "as is" without warranty of any kind. See the [Terms of Use](#) for more information.

Topic#: 002494

Created: August 2010

Introduction

This *Tech Note* outlines Microsoft's CPU optimization strategy for Windows 7 and Windows Server 2008 R2, and how Wonderware Application Server is affected.

It also contains guidelines for optimizing Wonderware Application Server performance on the relevant operating systems.

Application Versions

- Wonderware Application Server 3.1 SP3
- Wonderware InTouch 10.1 SP3
- Wonderware Historian 10.0 SP1

Reduced Power Consumption Features

Microsoft has implemented CPU optimizations for saving power in newer operating systems, such as on Windows 7 (32 and 64-bit) and Windows Server 2008 R2. The following statement is taken from the [Microsoft Web site](#):

Windows Server 2008 introduced a 'balanced' power policy, which monitors the utilization level of the processors on the server and dynamically adjusts the processor performance states to limit power to the needs of the workload. Windows Server 2008 R2 enhances this power saving feature by adding Core Parking and expanding on power-oriented Group Policy settings. Active Directory Domain Services Group Policy in Windows Server 2008 already gave administrators a certain amount of control over power management on client PCs. These capabilities are enhanced in Windows Server 2008 R2 and Windows® 7 to provide even more precise control in more deployment scenarios for even greater potential savings.

When the PC has a multi-core CPU, the balanced power feature can impact the performance of some Wonderware Application Server operations, such as Galaxy Load and Galaxy Import operations. For example, the performance of those operations degrades on Windows Server 2008 R2 for GR Load/Import operations, compared to the Windows Server 2008 64-bit operating system.

How to Run at High Performance

To obtain optimal performance, **disable** the power-saving/CPU-optimization features. Shell commands to disable the CPU optimizations are described in a Microsoft document titled [Performance Tuning Guidelines for Windows Server 2008 R2](#).

This document includes a section called **Choosing Server Hardware: Performance Considerations, Tuning Processor Power Management Parameters**. The section describes the various tuning parameters for better performance. Excerpts from this Microsoft document are included here and are shown in italics.

The following settings have shown performance improvement on Windows 7 and Windows Server 2008 R2.

Minimum and Maximum Processor Performance State

Processors change between performance states ("P-states") very quickly to match supply to demand, delivering performance where necessary and saving power when possible. If your server has specific high-performance or minimum-power-consumption requirements, you might consider configuring the Minimum or Maximum Processor Performance State parameter.

The values for both the Minimum and Maximum Processor Performance State parameters are expressed as a percentage of maximum processor frequency, with a value in the range 0 – 100.

If your server requires low latency, invariant frequency, or high performance, you might not want the processors switching to lower-performance states. For such a server, you can cap the minimum processor performance state at 100 percent by using the following commands:

```
Powercfg -setacvalueindex scheme_current sub_processor 893dee8e-2bef-41e0-89c6-b55d0929964c 100
Powercfg -setactive scheme_current
```

If your server requires lower power consumption, you might want to cap the processor performance state at a percentage of maximum. For example, you can restrict the processor to 75 percent of maximum by using the following commands:

```
Powercfg -setacvalueindex scheme_current sub_processor bc5038f7-23e0-4960-96da-33abaf5935ec 75
Powercfg -setactive scheme_current
```

Processor Performance Boost Policy

Intel Turbo Boost Technology is a feature that allows Intel processors to achieve additional performance when it is most useful (that is, at high system loads). However, this feature increases CPU core power consumption, so we configure Turbo Boost based on the power policy that is in use. Turbo Boost is enabled for High Performance power plans and disabled on Balanced and Power Saver plans for the current generation of processors. For future processors, this default setting might change depending on the power efficiency of such features. To use the Turbo Boost feature under the Balanced or Power Saver plans, you must configure the Processor Performance Boost Policy parameter.

The Processor Performance Boost Policy is a percentage value from 0 to 100. The default value of this parameter is 35 percent on Balanced and Power Saver plans. Any value lower than 51 disables Turbo mode. To enable Turbo Mode, set this value to 51 or higher.

The following commands set Processor Performance Boost Policy to 100 on the current power plan. Specify the policy by using a GUID string, as shown below:

```
Powercfg -setacvalueindex scheme_current sub_processor 45bcc044-d885-43e2-8605-ee0ec6e96b59 100
Powercfg -setactive scheme_current
```

*Note that you must run the **powercfg -setactive** command to enable the new settings. You do not need to reboot the server.*

Processor Performance Core Parking Maximum and Minimum Cores

Core parking is a new feature in Windows Server 2008 R2. The processor power management (PPM) engine and the scheduler work together to dynamically adjust the number of cores that are running threads. The PPM engine chooses a minimum number of cores on which threads will be scheduled. Cores that are chosen to be “parked” do not have any threads scheduled on them and they can drop into a lower power state. The remaining set of “unparked” cores are responsible for the entirety of the workload (with the exception of affinized work or directed interrupts). Core parking can increase power efficiency during lower usage periods on the server because parked cores can drop into a low-power state.

For most servers, the default core-parking behavior provides the optimum balance of throughput and power efficiency. If your server has specific core-parking requirements, you can control the number of cores available to park by using either the Processor Performance Core Parking Maximum Cores parameter or the Processor Performance Core Parking Minimum Cores parameter in Windows Server 2008 R2.

The values for these parameters are percentages in the range 0–100. The Maximum Cores parameter controls the maximum percentage of cores that can be unparked at any time, while the Minimum Cores parameter controls the minimum percentage of cores that can be unparked. To turn off core parking, set the minimum cores parked to 100 percent by using the following commands:

```
Powercfg -setacvalueindex scheme_current sub_processor bc5038f7-23e0-4960-96da-33abaf5935ec 100  
Powercfg -setactive scheme_current
```

To reduce the number of schedulable cores to 50 percent of the maximum count, set the Maximum Cores parameter to 50 as follows:

```
Powercfg -setacvalueindex scheme_current sub_processor bc5038f7-23e0-4960-96da-33abaf5935ec 50  
Powercfg -setactive scheme_current
```

Wonderware R&D and Product Development Teams

Tech Notes are published occasionally by Wonderware Technical Support. Publisher: Invensys Systems, Inc., 26561 Rancho Parkway South, Lake Forest, CA 92630. There is also technical information on our software products at [Wonderware Technical Support](#).

For technical support questions, send an e-mail to support@wonderware.com.

 [Back to top](#)

©2010 Invensys Systems, Inc. All rights reserved. No part of the material protected by this copyright may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying, recording, broadcasting, or by any information storage and retrieval system, without permission in writing from Invensys Systems, Inc.
[Terms of Use](#).