Registry Reflection in Windows

May 8, 2009

Abstract

Abstract

Registry reflection in the 64-bit version of Windows® 32-bit On Windows 64-bit (WOW64) copies specific registry keys and values between the 32-bit and 64-bit views of the registry to keep them synchronized, but conflicts can occur. This paper describes mechanisms for controlling registry reflection to prevent conflicts between 32-bit and 64-bit applications.

This information applies for the following operating systems:
 Windows Vista®
 Windows Server® 2008
 Windows Server 2003
 Windows XP Professional x64 Edition

The current version of this paper is maintained on the Web at:
 <http://www.microsoft.com/whdc/system/platform/64bit/RegReflect.mspx>

References and resources discussed here are listed at the end of this paper.

Disclaimer: This is a preliminary document and may be changed substantially prior to final commercial release of the software described herein.

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This White Paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, email address, logo, person, place or event is intended or should be inferred.

© 2009 Microsoft Corporation. All rights reserved.

Microsoft, Windows, Windows Server, and Windows Vista are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

Document History

| **Date** | **Change** |  |  |  |
| --- | --- | --- | --- | --- |
| May 8, 2009 | Update to indicate that Windows 7 does not support reflection and to clarify the rules for enabling/disabling reflection. |
| July 22, 2007 | Update to Resource link |
| August 30, 2005 | Original publication |

Contents

[Introduction 3](#_Toc229481582)

[Registry Reflection Behavior 3](#_Toc229481583)

[Registry Reflection in Windows Server 2003 4](#_Toc229481584)

[Registry Reflection Scenarios 5](#_Toc229481585)

[Registry Reflection Enabled 5](#_Toc229481586)

[Registry Reflection Disabled 5](#_Toc229481587)

[Next Steps 5](#_Toc229481588)

[Resources 6](#_Toc229481589)

# Introduction

The 64-bit version of Windows® 32-bit On Windows 64-bit (WOW64) provides a 32‑bit emulation layer on 64-bit versions of the Windows operating system. One of the primary responsibilities of WOW64 is to isolate 32-bit applications from 64-bit applications, which includes preventing file and registry collisions.

To prevent registry collisions, WOW64 presents 32-bit applications with an alternative view of the registry by splitting key portions of the registry (most importantly, HKLM\Software) into a 32-bit view and a 64-bit view.

Although having two views of HKLM\Software prevents collisions between 32-bit and 64-bit applications, these applications still must have a level of interoperability through COM and other mechanisms. Registry reflection copies specific registry keys and values between the two registry views to keep them synchronized. However, occasionally conflicts occur that prevent applications from running correctly.

This paper describes mechanisms for controlling registry reflection in Windows XP Professional x64 Edition, Windows Server® 2003 with SP1, Windows Vista®, and Windows Server 2008 to prevent conflicts between 32-bit and 64-bit applications.

**Note:** Registry reflection has been removed fromWindows 7, Windows Server 2008 R2, and later releases of Windows.

# Registry Reflection Behavior

Registry reflection was first introduced in Windows XP Professional x64 Edition as part of WOW64. Reflection was fully enabled by default and could not be disabled or modified.

The default registry reflection behavior depends on the order in which applications are installed and therefore the order in which registry keys are created:

* If a key is first created in one view of the registry, it is created by reflection in the other view of the registry. For example, if a key is created in the 32-bit registry, it is also created in the 64-bit registry.
* If a key that already exists in one view of the registry is created in the other view, the existing key's value is overwritten with the new registry key value by reflection. For example, if a key that already exists in the 64-bit registry is created in the 32-bit registry, then the 64-bit registry key value is overwritten with the new value in the 32-bit registry.

The reflection operation actually occurs when the handle to the open key is closed after an update operation. If the application never closes the handle, then the reflection operation occurs when the system cleans up during process termination.

Another way of describing registry reflection is “the last writer wins,” as illustrated by the following example:

1. An administrator installs the 32-bit version of Microsoft® Office, which registers the 32-bit version of Winword.exe to handle .docx files.

2. The registry reflector copies this information to the 64-bit registry view, which causes both 32-bit and 64-bit applications to launch the 32-bit version of Winword.exe for .docx files.

3. An administrator then installs the 64-bit version of Office, which registers the 64-bit version of Winword.exe to handle .docx files in the 64-bit registry view.

4. The registry reflector copies this information into the 32-bit registry, which overwrites the existing registry key value. As a result, both 32-bit and 64-bit applications launch the 64-bit version of Winword.exe for .docx files.

# Registry Reflection in Windows Server 2003

When 32-bit and 64-bit versions were installed on the same system, some application installations broke in some scenarios due to conflicts that occurred when registry keys were written or removed in a particular order. For example, suppose a user installed a 64-bit version of an application that created a particular registry key and then installed a 32-bit version that updated the key for both versions. If the user then removed the 32-bit version, its installer would delete the registry key, which would break the 64-bit version.

To prevent such conflicts, the RegDisableReflectionKey, RegEnableReflectionKey, and RegQueryReflectionKey functions were introduced in Windows Server 2003 with SP1. Installers and applications can use these functions to completely disable (or reenable) reflection for individual registry keys.

If reflection is disabled for a specific registry key, that key is not reflected under any circumstances, even when no conflict would occur. The behavior is as follows:

* If a key is created in one registry view, it is created only in that view and not in the other registry view. For example, if a key is created in the 32-bit registry view, it is not also created in the 64-bit registry view.
* If a key that already exists in one view of the registry is created in the other view, the existing key's value is preserved—it is not overwritten by the new registry key value. For example, if a key that already exists in the 64-bit registry is created in the 32-bit registry, the 64-bit registry key value is not changed.

The following rules apply to registry reflection:

* Installers and applications should disable reflection only for the registry keys that they create.
* If an application or installer tries to disable reflection on a key that is not reflected, RegDisableReflectionKey succeeds but has no effect.
* If an application or installer tries to enable reflection on a key that is not reflected by default, **RegEnableReflectionKey** succeeds but has no effect.

# Registry Reflection Scenarios

The examples in this section illustrate how registry reflection works for the following scenarios:

* Registry reflection enabled
* Registry reflection disabled

## Registry Reflection Enabled

When registry reflection is enabled, reflection is bidirectional between the 32-bit and 64-bit versions of the registry. For example, the following table shows the steps that occur when registry key “A” is created and edited.

| **32-bit registry** | **Reflection** | **64-bit registry** |
| --- | --- | --- |
| 1. Key A is created by an application installer. |  | 2. Key A is created by reflection. |
| 4. Key A is updated by reflection. |  | 3. Key A is edited by a 64-bit application. |
| 5. Key A is edited by a 32-bit application. |  | 6. Key A is updated by reflection. |

## Registry Reflection Disabled

Windows Server 2003 SP1 provides the ability to disable registry reflection on a specific key. When registry reflection is disabled for Key A, creations and edits in one view of the registry are not reflected in the other view, as shown in the following table.

| **32-bit registry** | **Reflection** | **64-bit registry** |
| --- | --- | --- |
| 1. Key A is created by a 32-bit application or installer. |  | 2. Key A is not created. |
| 4. Key A is not updated.  |  | 3. Key A is created by a 64-bit application or installer. |
| 5. Key A is edited by a 32-bit application or installer. |  | 6. Key A is not updated. |
| 8. Key A is not updated.  |  | 7. Key A is edited by a 64-bit application or installer. |

# Next Steps

Software vendors who build both 32-bit and 64-bit versions of an application should test their applications thoroughly for conflicts between 32-bit and 64-bit registry keys by:

* Installing both versions of the application on a single 64-bit system.
* Testing both versions of the application with registry reflection fully enabled to determine whether conflicts occur for any registry keys.
* If conflicts occur, disabling reflection to resolve the conflict as described in this paper.

# Resources

#### WHDC Web site

64-bit System Design nodal page

<http://www.microsoft.com/whdc/system/platform/64bit/default.mspx>

#### Windows Driver Kit

INF AddReg Directive

<http://msdn.microsoft.com/en-us/library/ms794514.aspx>

#### MSDN

Registry Redirector

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/winprog64/winprog64/registry_redirector.asp>

Registry Reflection

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/winprog64/winprog64/registry_reflection.asp>