Managing the System Registry Hive on Windows Server 2003 and Windows Server 2008 Integrity Systems

Published: November 2009
Edition: 1.0, Part Number: 5992-5670

Table of Contents

List of Figures

About This Document

This white paper provides an explanation of the system registry hive and its limitations, and gives some methods for managing those limitations. The opinions in this document may suit some environments more than others.

Intended Audience

This document is intended for system architects and administrators responsible for managing HP Integrity servers. Architects and administrators are expected to know operating system concepts, commands, and configuration, as well as networking concepts and configuration. Some manipulation of the system registry file is necessary, so a familiarity with the Registry Editor application (RegEdit) is required. You should also be able to liaise with different departments in your organization, such as the Storage Area Networks (SAN) group, specialist application groups, and Change Management/Documentation departments.

This document is not a tutorial.

Typographic Conventions

This document uses the following typographical conventions:

`%`, `$`, or `#`	A percent sign represents the C shell system prompt. A dollar sign represents the system prompt for the Bourne, Korn, and POSIX shells. A number sign represents the superuser prompt.
`Command`	A command name or qualified command phrase.
`Computer output`	Text displayed by the computer.
Ctrl+x	A key sequence. A sequence such as **Ctrl+x** indicates that you must hold down the key labeled **Ctrl** while you press another key or mouse button.
Key	The name of a keyboard key. **Return** and **Enter** both refer to the same key.
`User input`	Commands and other text that you type.
`Variable`	The name of a placeholder in a command, function, or other syntax display that you replace with an actual value.
WARNING	A warning calls attention to important information that if not understood or followed will result in personal injury or nonrecoverable system problems.
CAUTION	A caution calls attention to important information that if not understood or followed will result in data loss, data corruption, or damage to hardware or software.
IMPORTANT	This alert provides essential information to explain a concept or to complete a task
NOTE	A note contains additional information to emphasize or supplement important points of the main text.

Technical Review

Introduction

The Windows Server 2003 and Windows Server 2008 operating systems have a system registry hive size limitation of 32 MB (for all Service Packs). For most customer configurations, it is unlikely this limit will ever be reached. But in certain circumstances, particularly for customers with large configurations or numbers of SAN disks using multipath MPIO technology, the 32 MB limit is reachable, and can cause the system to fail to boot when restarted.

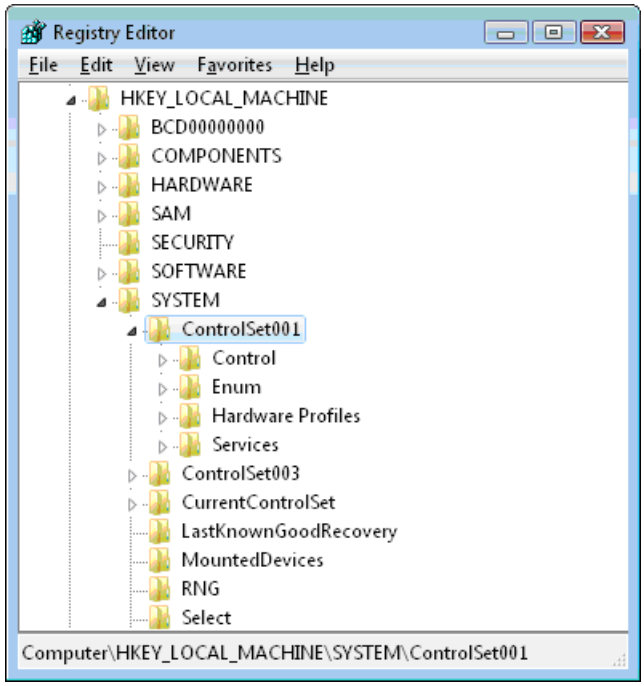
HP and Microsoft have worked together to create methods of recovery, proactive notification, and design avoidance should the System hive approach its 32 MB limit. By using proactive notification and the other methods described in this white paper, reboot failures of this type can be avoided.

Understanding System Hive Fundamentals and Limits

The System hive is a critical registry component. It is referenced by the Registry Editor application (`RegEdit.exe`, or RegEdit) as "SYSTEM". Below the System hive, structurally, are a number of ControlSets. The origin of these ControlSets goes back to the Windows NT operating system. A ControlSet encapsulates all of the system's hardware and software information, such as device driver and SAN volume details, in a tree structure. Because this data is critical to system operation, anytime the System hive is changed by adding a new hardware component or software package, a new ControlSet is created.

Microsoft designed the system to maintain more than one ControlSet in order to achieve system resilience. Typically there are two ControlSets, such as ControlSet001 and ControlSet002. The system rotates between these two ControlSets every time a change is made. This functionality allows an Administrator to revert back to the "Last Known Good Configuration" (LastKnownGood, or LKG) by pressing the **F8** key should the system fail to boot. Figure 1 shows a typical System hive ControlSet, as displayed by RegEdit.

Figure 1 System Hive ControlSet structure

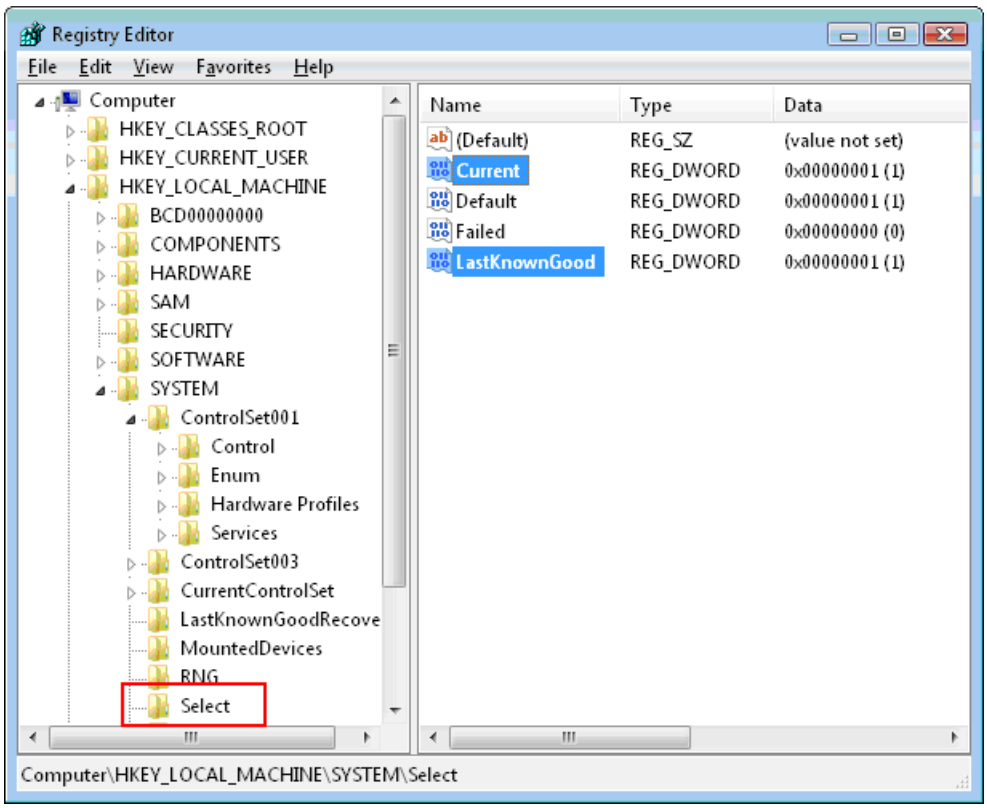


Note that Figure 1 shows ControlSet001 and ControlSet003. These numbers do not have to be contiguous. There can even be more than two ControlSets in extreme circumstances, such as a

known failed ControlSet. For the purposes of this document, the CurrentControlSet key is considered an exact replica or alias of one of the other ControlSets, and does not consume any System hive space.

The other important key to note is the Select key, as shown in Figure 2.

Figure 2 Select Key values



In the Select key, the Current and LastKnownGood subkey values are critically important. In Figure 2 you can see the values for both are "1". If an application or hardware change had recently occurred, the LastKnownGood value would be different than the Current value. Therefore, in the example above, this means that ControlSet003 is actually superfluous and could theoretically be deleted in order to gain space in the System hive.

IMPORTANT: If the system is running Windows Server 2003 for Itanium-based Systems, an understanding of Select key values is critical for configuring the system to reduce the size of the System hive.

Estimating System Hive Size

The System hive is essentially a database of keys that is stored on the filesystem of the boot drive in the following location:

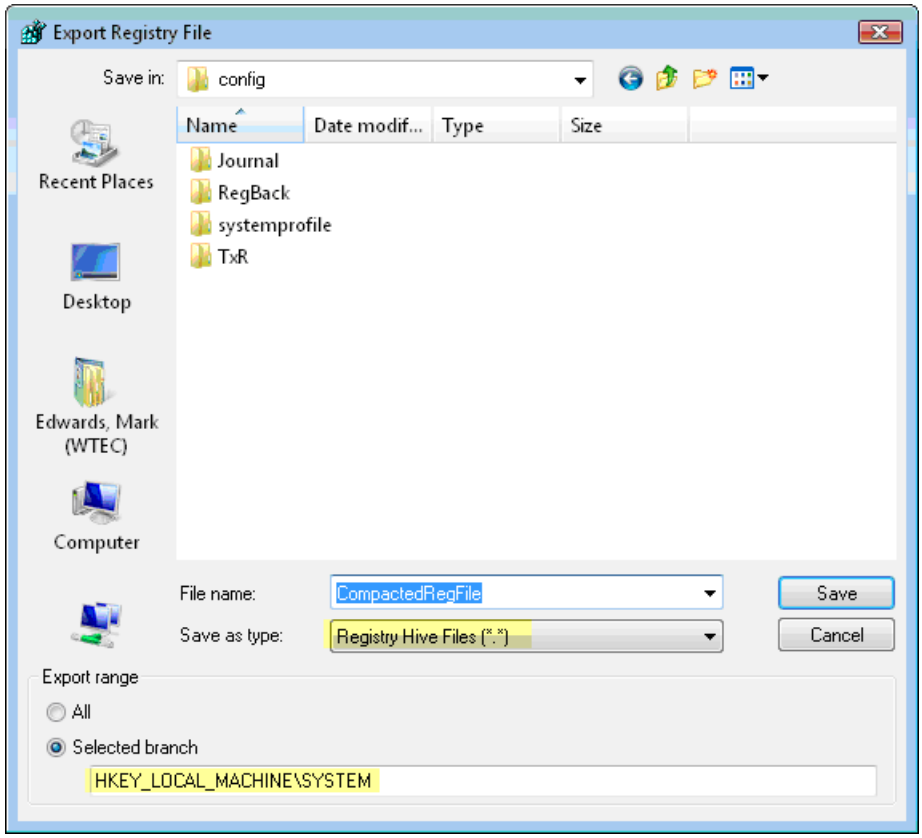
`%SystemRoot%\System32\Config\SYSTEM`

It is important to note that the size of the file does not necessarily reflect the size of that database. This is because the System hive never shrinks under normal operation. For example, when 5 MB of data is added to the System hive, its size grows by 5 MB (to a maximum of 32 MB). However, if that same 5 MB of data is deleted from the hive, its file size does not change, even though that space is now available for reuse. Therefore, a hive size of 32 MB does not necessarily mean that much space is being utilized, and normal system operation can still proceed.

The easiest method to determine true hive size is to use RegEdit to save the System hive in binary form. RegEdit has the ability to compact the System hive and remove any free space, thus reporting

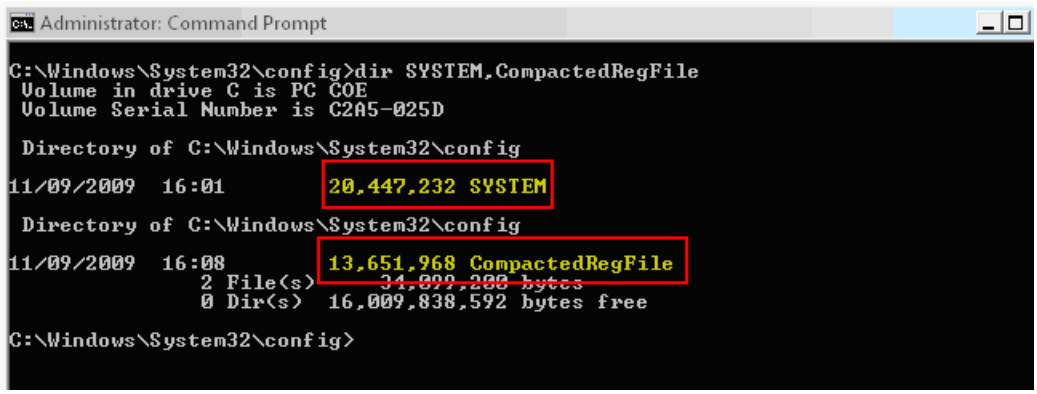
its true size. Figure 3 shows the correct options for saving the System hive. Note the highlighted fields.

Figure 3 RegEdit Export screen



In the example above, a new file is created called `CompactedRegFile`. Now, by examining the file system, the size difference between the `SYSTEM` and `CompactedRegFile` files can be determined. Figure 4 shows the output of a directory listing.

Figure 4 Directory listing showing difference in file size



The difference in this example is approximately 7 MB, indicating that roughly 35% of the System hive is not being used. This unused space is consumed prior to any expansion of the actual file size, so in the example above, 7 MB of data must be added to the System hive before any increase in file size becomes apparent.

In the section, "Proactive Avoidance" (page 20), there is a discussion of an HP Management Service that monitors and alerts Administrators when the System hive is approaching its size limit. That service leverages existing SNMP- or WBEM-based management infrastructure.

Causes of Increasing System Hive Size

When Windows Server 2003 for Itanium-based Systems was first released, the System hive limit was set to 32 MB by design. This was not changed with the release of Windows Server 2008 for Itanium-based Systems (although it was increased in Windows Server 2008 R2 for Itanium-based Systems). At the inception of Windows Server 2003, Microsoft also introduced a new technology called Multipath I/O (MPIO). This was Microsoft's strategy for disk multipathing, and included software that vendors could plug into. As multipathing in storage area networks (SANs) became more pervasive, the information to be managed (about each disk and path) grew in the System hive. Up until March 2009, the MPIO framework allowed a maximum of 8 paths per disk. So if there were 100 disks, then information about 800 different paths had to be stored in each ControlSet. With the March release of the MPIO framework (Microsoft internal version 1.22, which the major manufacturers have built their modules on), the maximum number of paths per disk is now 32, thus quadrupling the space required in each ControlSet for disk information.

Compounding the problem, a popular application often used on Itanium-based scale-up systems is Symantec's Veritas Storage Foundation for Windows (SFW). This application is generally used when large quantities of disks need to be managed. For each disk managed by SFW, an additional entry is added to the ControlSet.

Another reason for increased System hive size is when there are more than two ControlSets. This is rare. It only happens when the primary ControlSet failed to load and a "LastKnownGood" configuration had to be used. When it occurs, the System hive Select key will have a value other than "0" (zero) in its Failed subkey (refer back to Figure 2). The size of a System hive containing a failed ControlSet is increased by approximately a third. This can be advantageous however, since the first two contributing causes listed above tend to increment slowly over time, making it difficult to determine the cause, whereas a ControlSet failure is much more sudden and easier to root-cause.

There are other secondary causes for increased System hive size, but they are only manifestations of the primary causes listed above. Noteworthy of these would be a SAN administrator inadvertently adding a number of disks to a system incorrectly, and then unpresenting them. In this situation, Windows maintains a record of these entries since it cannot determine if they are stale or transient (for example, transient disks would be shared cluster disks), and then adds them to the ControlSet, never to be removed. A similar situation can occur when a SAN device's firmware changes its identity string. Typically the firmware revision, or an incantation of it, is appended to the Hardware ID subkey. If there are a large number of disks, a firmware update can be the catalyst that causes a breach of the 32 MB System hive limit.

Breaching the System Hive Limit

As already noted, the System hive limit is hard-coded to 32 MB. This limit cannot be breached. The file is mapped into system memory with that restriction. This means that whenever a large addition is made to the system registry, one that takes it to the 32 MB limit, any "overflow" is lost. In this scenario the registry reaches a point where nothing new can be added, so applications expecting to add keys cannot function and fail.

Because it is hard-coded, there is no possible way to bypass this system limitation without a radical redesign of the operating system. In sections, "System Recovery" (page 11), and "Proactive Avoidance" (page 20), are discussions on how to recover should the server ever breach the limit, and how to avoid reaching the limit proactively.

Differences in Windows Server 2003, Windows Server 2008, and Windows Server 2008 R2

Windows Server 2003 and Windows Server 2008 for Itanium-based Systems have the same 32 MB limitation for all Service Pack versions. Both operating systems typically collect the same data in the System hive and are therefore subject to the same root causes of increased hive size. Windows Server 2008 does have a specific hotfix to alleviate the symptoms, which is described

in the Recovery section of this document. While no such hotfix exists for Windows Server 2003, the same result can also be achieved for that operating system.

Of these two operating systems, the hive limit is more likely reached in those running Windows Server 2008, mainly because these are newer installations and tend to run the latest software products. For example, the MPIO framework discussed earlier is more likely the newer version supporting 32 device paths, thus increasing System hive size considerably. Windows Server 2003 systems often have the older version of MPIO installed, the one supporting 8 paths, and so exert less pressure on hive size. However, be aware that any new installation, Windows Server 2003 or Windows Server 2008, can take the latest MPIO framework, and is therefore susceptible to breaching the limit.

In Windows Server 2008 R2, Microsoft raised the System hive limit considerably for the x64 and Itanium editions. Both architectures now set their limit to one of the following, whichever is less:

- 1.5 Terabytes
- One half of physical memory

In addition, these barriers are now arbitrary in R2 and can be raised much more easily, due to a redesign of the boot process.

System Recovery

General Outlines for Recovery

Recovery is needed whenever the System hive reaches its limit, meaning the `SYSTEM` file size is 32 MB exactly, and there is no reusable space in the file. When this happens, as the operating system boots, it cannot copy the entire hive without truncating it, and usually fails with a Stop Code of 0x7B. Most administrators are familiar with Stop Code 0x7B since it is reserved for inaccessible boot devices. However, a SYSTEM file size of 32 MB with no reusable space also causes it.

To recover, space must be created in the System hive so the system can boot. The method for doing this is to boot the system to either WinPE (Windows Preinstallation Environment) or WinRE (Windows Recovery Environment, if installed).

NOTE: For the remainder of this document, WinPE and WinRE are considered equals from a procedural standpoint, so whenever WinPE is mentioned, remember that the same tasks can be achieved with WinRE.

WinPE is the most suitable recovery environment since any RegEdit deletions are done in the context of the System Account, which can delete much more than a normal Administrator account (without taking ownership of the subkeys). From WinPE the hive must be manipulated to create space at boot in order to get past the 0x7B Stop Code. The next few sections explain how to do this for both operating systems.

In Windows Server 2008 the recovery process is simpler because there are two hotfixes that can be applied. Even so, space must still be created in the System hive first, in order to boot the operating system past the Stop Code, since Hotfixes cannot be installed using WinPE.

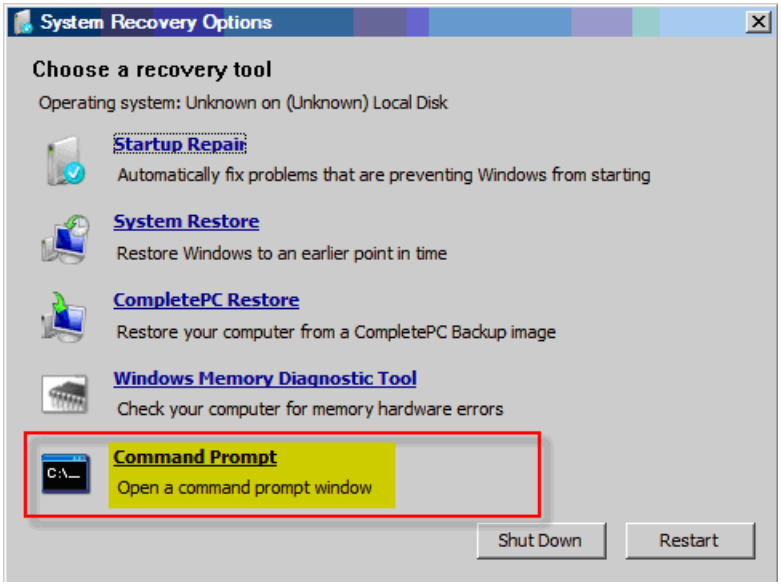
Booting WinPE/WinRE and Loading the System Hive Registry

The method used for booting to the recovery environment will depend on whether the system was purchased with Windows pre-loaded at the factory, by HP, or if Windows was purchased from Microsoft. HP Integrity servers that come pre-installed with Windows include an HP Reinstall DVD. This media boots to WinPE and provides a recovery shell. If Windows was purchased from Microsoft, through a Volume License agreement for example, then *that* DVD image must be booted and the **F8** key pressed to get to the Windows Recovery Environment.

Once the appropriate shell loads, a Command Prompt option becomes available. Figure 5 shows this option in the Windows Preinstallation Environment (the HP Reinstall Media DVD), while Figure 6 shows it in the Windows Recovery Environment.

Figure 5 Command Prompt option in WinPE (HP Reinstall media)

Figure 6 Command Prompt option in WinRE



At this point, to load the system registry, complete the following steps:

1. At the command prompt, type `regedit` and press **Enter** to run RegEdit.

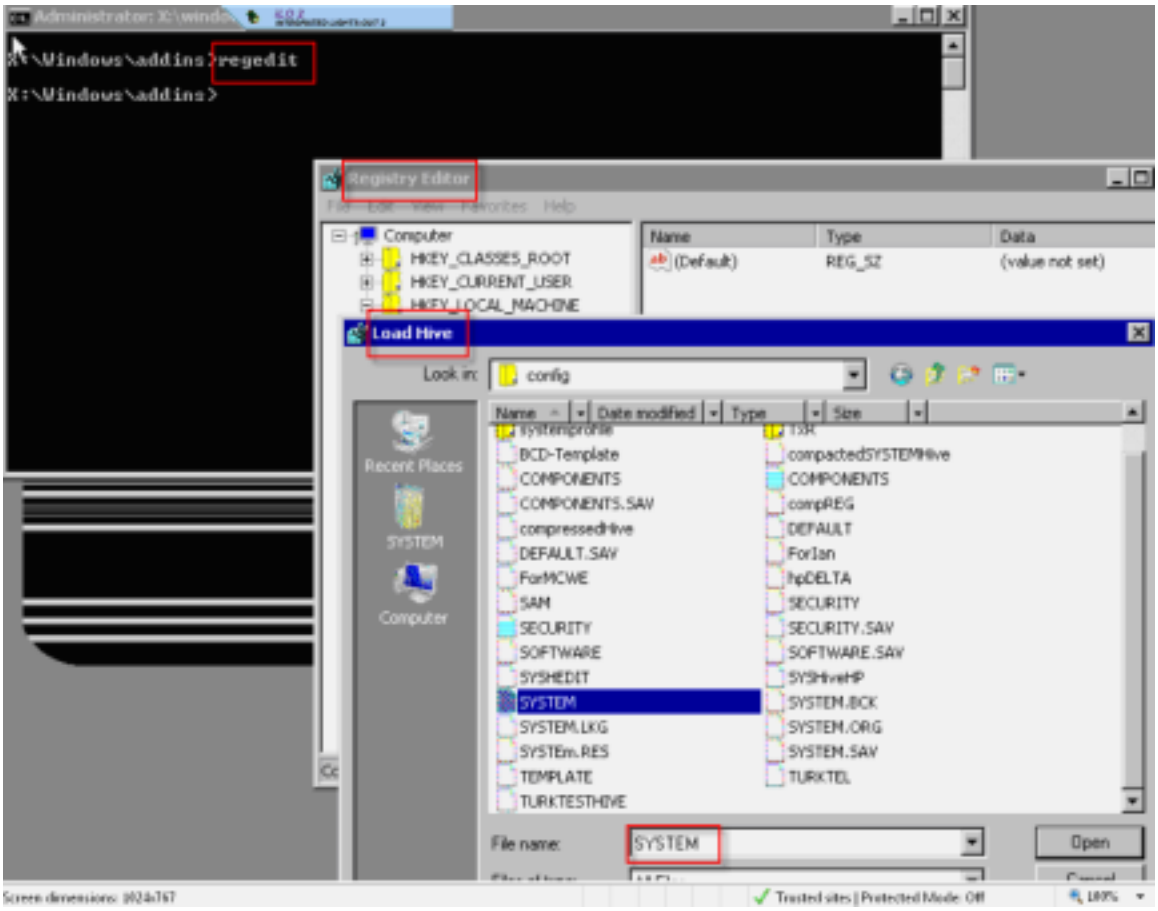
NOTE: This does not load the desired system registry hive, but the registry of the WinPE environment instead.

2. To load the system registry hive, select the HKEY_LOCAL_MACHINE key in the left pane of the RegEdit window by clicking on it. This should highlight that key only.
3. Select **File** → **Load Hive**. If the Load Hive option is greyed out, then the HKEY_LOCAL_MACHINE key was not highlighted correctly in the previous step.

   Since WinPE does not know which drive is associated with which letter (like the operating system does), it is possible that the system drive is not listed as the C: drive. However, the volume labels are displayed, so figuring out the correct drive should not be difficult.
4. Once the correct drive is determined it is wise to create a backup of the `SYSTEM` file. Use the command prompt (OK to use while RegEdit is up) to select the drive where the system files are located (usually the C: drive). Use the following command to create a copy of the System hive named `SYSTEM.BCK` (replacing the "C:" drive letter with a different one, if necessary):

   ```
   copy c:\windows\system32\config\SYSTEM c:\windows\system32\config\
   SYSTEM.BCK
   ```
5. Now that a backup exists, the original System hive must be loaded into the WinPE registry. Since WinPE already has its own System hive, a name must be selected for the real System hive, so it can be listed. For example `EditSYS` can be used. Figure 7 shows the loading of the System hive for a system in the WinPE environment (using HP Reinstall Media), where the "Advanced Command Prompt" button has been selected.

Figure 7 Loading the System Hive in WinPE (HP Reinstall media)



Once the target System hive registry is loaded, the method for recovery is different, depending on the operating system:

- For Windows Server 2003, see the section: "Recovery Specifics: Windows Server 2003" (page 14).
- For Windows Server 2008, see the section: " Recovery Specifics: Windows Server 2008" (page 15).

Recovery Specifics: Windows Server 2003

Recovery in Windows Server 2003 is slightly different from Windows Server 2008, given there is no hotfix to assist. Some of the required registry deletions that are automated by the Windows Server 2008 hotfix must be handled manually in Windows Server 2003.

For recovery, the approach taken here is to configure the system so that it only manages a single ControlSet. This method frees up considerable space in the registry. Later, after the operating system has booted, a pseudo "LastKnownGood" registry is created should it ever be needed for system recovery (refer to "Creating a Pseudo LastKnownGood System Registry Hive for Windows Server 2003 and Windows Server 2008" (page 16) for instructions).

To prevent the operating system from managing multiple ControlSets, a registry subkey (ReportBootOK) must be edited to disable the feature. More details about this registry subkey are found here:

http://technet.microsoft.com/en-us/library/cc739989(WS.10).aspx

To implement this approach, complete the following steps:

1. The ReportBootOK subkey resides in the Software hive, which means another hive must be loaded using RegEdit. Load the Software hive in the same manner that you loaded the System hive in the previous section. As before, a name for the Software hive must be given while loaded. As an example, `EditSOFTWARE` can be used.
2. Once the EditSOFTWARE hive is loaded, enumerate the following keys:

   `HKLM\EditSOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon`

   In the Winlogon key is the value named ReportBootOK. By default this value is set to "1". You must change this value to "0". This prevents the creation of more ControlSets.
3. Unload the EditSOFTWARE hive by highlighting it, then selecting **File → Unload Hive**, which automatically writes the changes to the system Software hive.
4. Delete the duplicate ControlSet(s). This is done by viewing the server's System hive Select key (created as EditSYS in the previous section). Figure 2 (page 8) shows an example of the values in the Select key. The value of interest is the Current value. This value will be a single digit, which denotes the ControlSet that the system boots from. For example, if the Current value is set to "1" this indicates ControlSet001 is the ControlSet the system boots from.

   The ControlSet that the system boots from must always be regarded as the "golden" ControlSet, and should never be deleted. However, the other ControlSet(s) in the server's System hive can now be deleted. For example, if the Current ControlSet is "1", and ControlSet001 and ControlSet003 are listed in the System hive, then ControlSet003 can be deleted.

   Note that the Registry Editor under WinPE may not have sufficient rights to delete all keys of the ControlSet (since it impersonates the System Account). If this is the case, the system will still boot, but will not have all of the other ControlSets deleted. The remainder of a partially-deleted ControlSet can be deleted once the system is back up after recovery is completed. The primary goal here is to create some space in the System hive, so the system will boot.
5. Once the unnecessary ControlSet(s) have been deleted (or partially deleted), then the EditSYS hive must be unloaded from the Registry Editor. Do this by highlighting the EditSYS hive

and selecting **File** → **Unload Hive**. When the Hive unloads, all changes are automatically saved.

6. Reboot the system. Further deletion of any redundant ControlSet(s) may be necessary at this time. If removal of unnecessary ControlSet(s) is difficult due to complex permissions, then contact Microsoft Support Services, since they can assist in deleting them fully. Since the system is now managing only one ControlSet (despite a second one existing), there will be no additions to the unused ControlSet. This means the server can be rebooted without risk of the 0x7b Stop Code, as long as system hive increases were minimal.
7. Perform the steps described in the section, "Creating a Pseudo LastKnownGood System Registry Hive for Windows Server 2003 and Windows Server 2008" (page 16). Also carefully review all of the section, "Proactive Avoidance" (page 20), to avoid the problem in the future.

Recovery Specifics: Windows Server 2008

Recovery in Windows Server 2008 is subtly different, as there are two hotfixes that must be installed. But since the server is not booting due to Stop Code 0x7B, some space must be created first in the current System hive before the hotfixes can be installed. The approach taken here is once again (as with Windows Server 2003) to create a System hive with a single ControlSet, thus freeing up considerable space in the registry. Later, after the operating system boots, a pseudo LastKnownGood registry is created should it ever be needed for system recovery (refer to "Creating a Pseudo LastKnownGood System Registry Hive for Windows Server 2003 and Windows Server 2008" (page 16) for instructions).

To implement this approach, complete the following steps:

1. Delete the duplicate ControlSet(s). This is done by viewing the server's System hive Select key (created as EditSYS in an earlier section). Figure 2 (page 8) shows an example of the values in the Select key. The value of interest is the Current value. This value will be a single digit, which denotes the ControlSet that the system boots from. For example, if the Current value is set to "1" this indicates ControlSet001 is the ControlSet the system boots from.

   The ControlSet that the system boots from must always be regarded as the "golden" ControlSet, and should never be deleted. However, the other ControlSet(s) in the server's System hive can now be deleted. For example, if the Current ControlSet is "1", and ControlSet001 and ControlSet003 are listed in the System hive, then ControlSet003 can be deleted.

   Note that the Registry Editor under WinPE may not have sufficient rights to delete all keys of the ControlSet (since it impersonates the System Account). If this is the case, the system will still boot, but will not have all of the other ControlSets deleted. The remainder of a partially-deleted ControlSet can be deleted once the system is back up after recovery is completed. The primary goal here is to create some space in the System hive, so the system will boot.

2. Once the unnecessary ControlSet(s) have been deleted (or partially deleted), then the EditSYS hive must be unloaded from the Registry Editor. Do this by highlighting the EditSYS hive and selecting **File** → **Unload Hive**. When the Hive unloads, all changes are automatically saved.
3. Reboot the system.
4. When the system boots back to the operating system level, contact Microsoft to obtain the following two hotfixes: KB973816 and KB973817. Details of these hotfixes are found here:

   http://support.microsoft.com/kb/973816

   http://support.microsoft.com/kb/973817

   Hotfix KB973816 prevents the server from creating another ControlSet, as well as deletes all of the other ControlSets from System hive. This hotfix requires a reboot, and directs you to add a value to the registry to enable this functionality.

Hotfix KB973817 provides a replacement for `Reg.exe` that enables compression of the hive and allows the hive's true size to be ascertained from the command line (the graphical utility RegEdit already has this functionality).

NOTE: Neither of these hotfixes can be installed on systems running Windows Server 2003 or Windows Server 2008 R2.

5. Perform the steps described in the section, "Creating a Pseudo LastKnownGood System Registry Hive for Windows Server 2003 and Windows Server 2008" (page 16). Also carefully review all of the section, "Proactive Avoidance" (page 20), to avoid the problem in the future.

Creating a Pseudo LastKnownGood System Registry Hive for Windows Server 2003 and Windows Server 2008

The recovery procedures outlined in previous sections will remove LastKnownGood functionality from the system. While rarely needed in modern servers, a pseudo LastKnownGood system registry hive can still be created every time the system boots, should the need for one arise. This involves scheduling a task to save the System hive to another location sixty seconds after boot, which requires a batch task.

To create this batch task, complete the following steps:

1. Use Notepad.exe or a similar text editor to create a `\Windows\System32\lkg.cmd` file with the following contents:

   ```
   move /y %windir%\system32\config\system.lkg %windir%\system32\config\system.plkg reg save hklm\system %windir%\system32\config\system.lkg /c /y
   ```

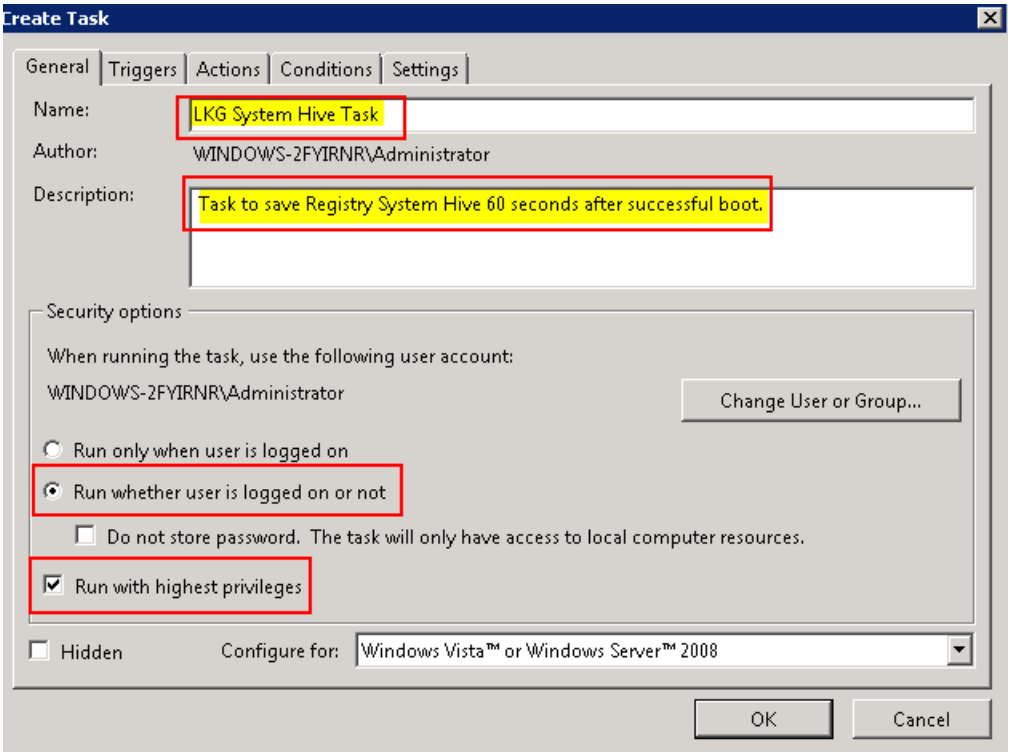
   This batch command creates a `system.lkg` file, which is a copy of the current hive after successful boot. Two copies of it are kept, `system.lkg` (current) and `system.plkg` (past).

2. Create the Scheduled Task by going to **Administrative Tools** → **Task Scheduler**.
3. In the General tab, change the following settings:
   - **Name:** change to "LKG system hive task"
   - **Description:** change to "Task to save Registry System Hive 60 seconds after successful boot."

   Also select the following check boxes:
   - Run whether user is logged on or not
   - Run with highest privileges (select only if running Windows Server 2008)

   When finished, the General tab should look like this:

Figure 8 General tab



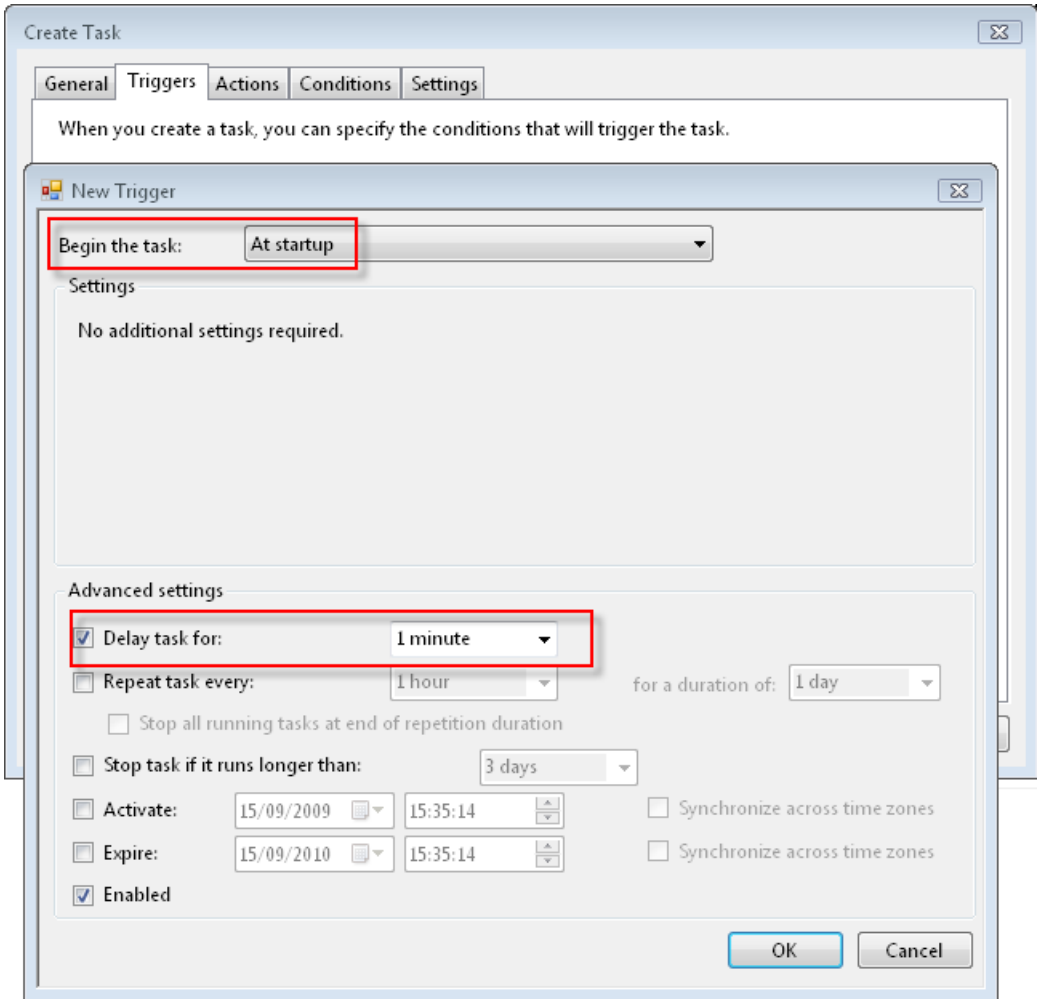
4. In the Triggers tab, change the following settings:
   - **Begin the task:** change to "At startup"

   Also select the following check boxes:
   - Delay Task for: 1 Minute

   The Triggers tab is not available on the Windows Server 2003 Task Scheduler, in which case the script must be run at startup. The task is delayed for 60 seconds in Windows Server 2008 to give the system more time to get into a running state. This is due to RegSave acting like the Export Hive function in RegEdit, where the hive will be locked while it is saved. Given that there are normally a large number of processes starting at boot time, delaying the Save by 60 seconds lets those finish.

   When finished, the Triggers tab should look like this:

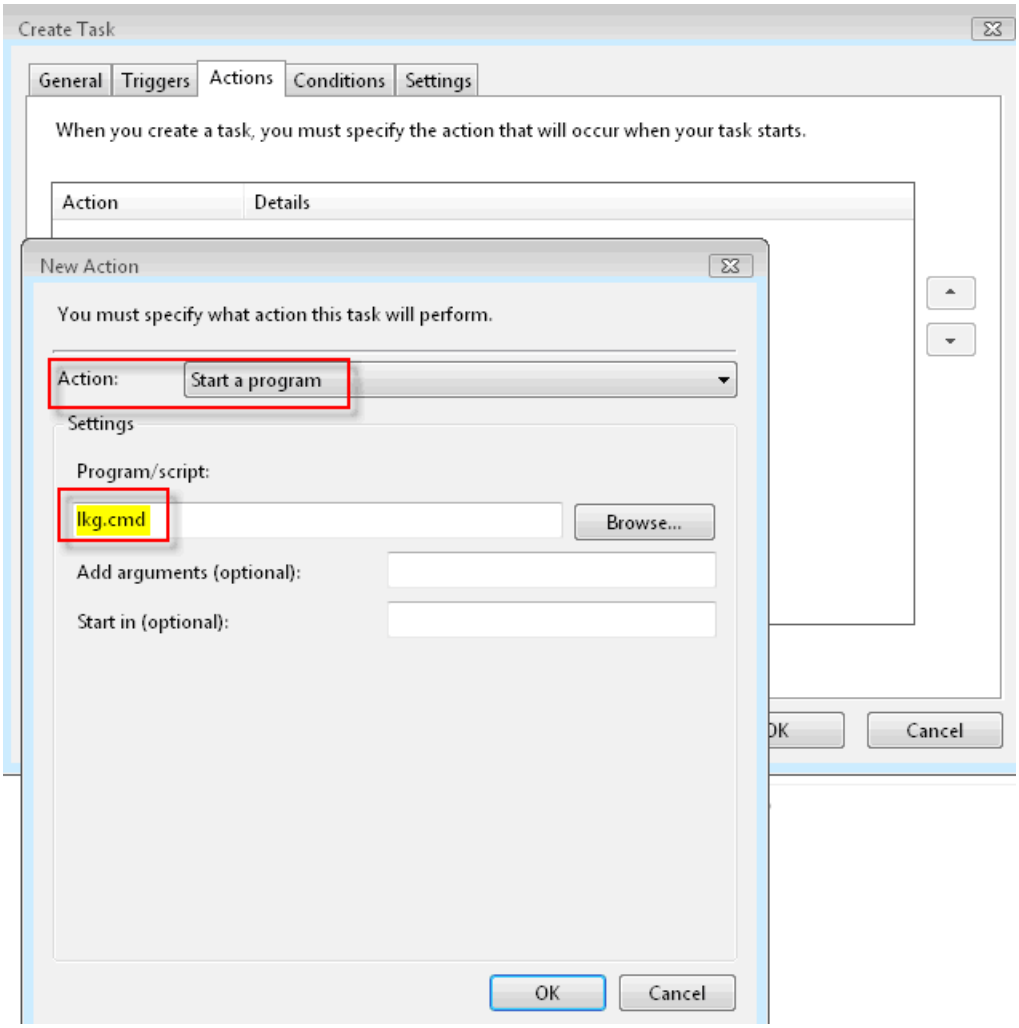
Figure 9 Triggers tab



5. In the Actions tab, change the following settings:
   - **Action:** change to “Start a program”
   - **Program/script:** change to “lkg.cmd”

   When finished, the Actions tab should look like this:

Figure 10 Actions tab



Administrator credentials must be supplied when the Task is added. No reboot is necessary for this task and it will take effect at the next reboot. To confirm the settings and script were created properly, perform a test reboot and confirm that the file `system.lkg` exists.

Using the Created Pseudo LastKnownGood (LKG)

Since there is no LastKnownGood ControlSet in the System hive anymore, the usual method of invoking it with the **F8** key no longer applies. Instead, WinPE must be booted to the Advanced Command Prompt (described earlier, see “Booting WinPE/WinRE and Loading the System Hive Registry” (page 11)). Then, copy the pseudo LastKnownGood created in the previous section over the System hive. It is good practice to make a backup of the current System hive before copying over it. The commands that accomplish this are:

```
rename c:\windows\system32\config\system c:\windows\system32\config\
system.fail
```

```
copy c:\windows\system32\config\system.lkg c:\windows\system32\config\
system
```

The server can now be rebooted without user intervention.

Proactive Avoidance

This section provides information on how to avoid the System hive 32 MB limit as well as how to install the HP Registry Monitor Service, which provides alerts when the limit is near.

Checking Disk Infrastructure

Typically the biggest contributor to hive usage is the amount of disks and number of paths to those disks. Administrators should know how many disks and paths a system has, and it's also a good idea to check how many are in the Registry, since a number of stale paths may exist. These are not removed when disks are deleted.

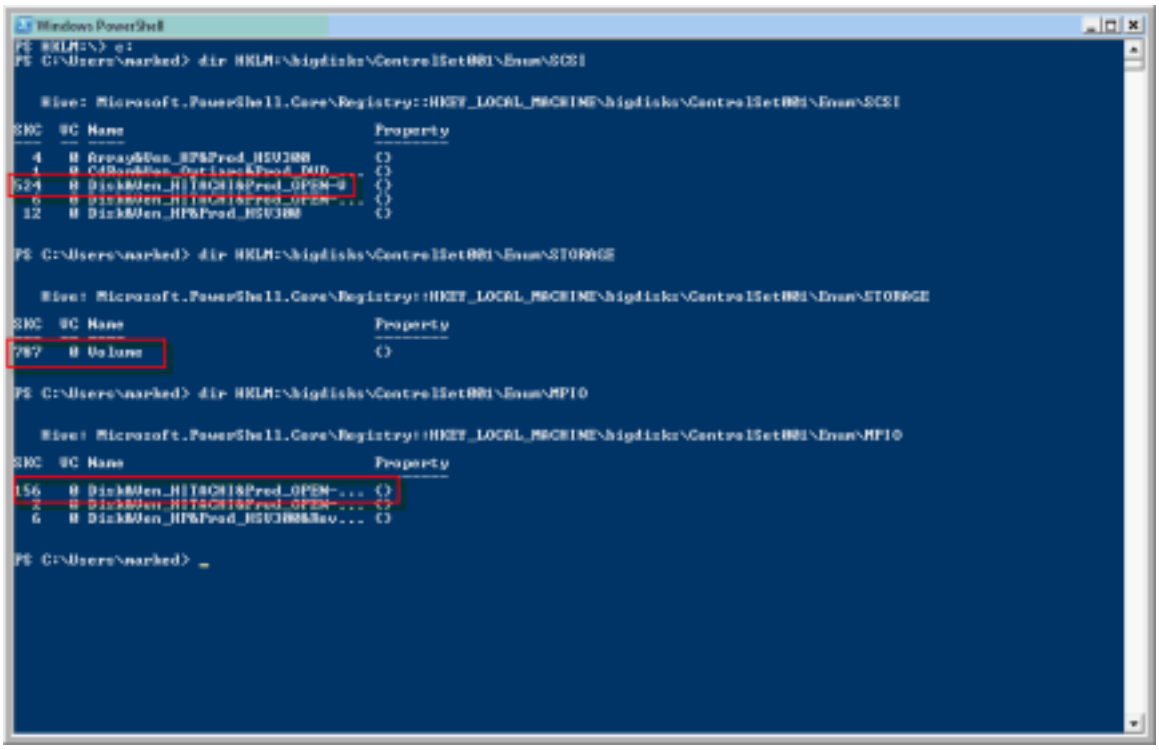
The best way to check the amount of registry entries is with a few simple commands in PowerShell. PowerShell has Registry Provider, which displays the registry in a folder structure. Administrators are used to navigating with the `cd` and `dir` commands, as well as the **Tab** key for folder completion. The Registry is indicated as "HKLM:", where the colon serves to make it act like a disk. From PowerShell, the command `dir hklm:\system` lists all of the subkeys, such as the ControlSets.

Using this information, it is possible to see how many disks the system recognizes by entering the following three commands:

```
dir hklm:\System\ControlSet001\Enum\SCSI
dir hklm:\System\ControlSet001\Enum\STORAGE
dir hklm:\System\ControlSet001\Enum\MPIO
```

Figure 11 is an example of the output of these commands, with PowerShell finding a large number of disks:

Figure 11 PowerShell screen



The output above shows there are currently 156 drives in the database (the MPIO) key, but the amount of SCSI disks recognized is 524, with a total of 787 Volumes. The MPIO key is the most useful since it only records one instance of the path to a disk. This means the Administrator can expect to see 156 drives (plus any local disks). The fact there are 524 SCSI disks is merely a representation of the MPIO disks with all of the paths enumerated. A cursory glance shows that

an average of approximately three to four paths are presented to each disk. The reality is that some disks may have 32 paths and some may have only 2, giving an average that is not very indicative.

What should be very clear here is that for 156 drives to have 787 volumes, something is definitely wrong. Either a number of disks have been deleted and represented, or another program is creating volumes. Reviewing the next section should give more insight into the number of volumes if Veritas Volume Manager (SFW) is installed on the system.

With this knowledge (the number of actual system drives, paths, and volumes), the next step is to perform a careful review of the registry entries to determine which entries are stale. For each ControlSet in the System hive, these stale drives, paths, and volumes should be deleted in order to reclaim a significant amount of System hive space.

Special Consideration for Symantec Veritas Volume Manager (SFW)

If the system uses Symantec Veritas Volume Manager, it would be a good idea to check how many volumes are being managed by it. This is done with the following command from a PowerShell prompt:

```
($VXVMCount = dir
hklm:\System\ControlSet001\Enum\STORAGE\Volumes\VXVM*).count
```

This gives the number of Veritas-managed volumes. If this number is larger than the volumes known to exist, Veritas provide a utility called VxScrub that can delete stale Windows and Veritas volumes. Please consult the Veritas Volume Manager documentation for use of VxScrub. While it is a very effective tool, VxScrub does require a mandatory reboot immediately after it deletes the stale entries.

Working with the SAN Administrator for MPIO Optimizations and SAN Maintenance

Most organizations split their System Administrator and SAN teams. This sometimes leads to miscommunication. If System hive space is near maximum, then any changes to disks or SAN devices can lead to additions to the registry without the Administrator's knowledge. A good example of this is the SCSI key, which in the previous example showed 524 paths. If all of these paths belonged to one SAN device, and the firmware was upgraded, giving a new identity string (such as a new firmware level appended to it), then all 524 paths would be replicated. Given this knowledge, you can see why it is so important for Administrators to know about any changes to the SAN infrastructure.

The number of paths to a disk may also surprise an Administrator. The MPIO Framework automatically enables every path to a maximum of 32, even though this might not be desired. Once the amount of paths is greater than 2, the additional paths are added to increase throughput to the device. In such cases it is worthwhile to analyze how many paths are needed for both redundancy and throughput. Given that current storage devices can easily have 4 or 8 ports, as well as the server having just two dual-port Fibre Channel host bus adapters, it is now possible for MPIO configurations to reach 32 paths easily. The question Administrators must ask is whether 32 paths are really necessary, since a reduction in paths will reduce the number of entries in the System hive, and therefore alleviate pressure on the limit.

Introducing the HP Registry Monitor Service

One of the problems with monitoring the System hive without automation is that its true size can differ from its indicated size. As discussed earlier, whenever elements are deleted, like an entire ControlSet, that space is marked as free inside the hive, but the size on disk remains the same. This means the indicated size of the file on disk could be just shy of 32 MB, which may worry an Administrator, but its true size could be closer to 12 or 13 MB. Since the true size of the consumed space cannot be gleaned with normal Windows tools, Hewlett-Packard has created a Management Service that monitors the true size of the System hive, sends an SNMP or WBEM

event, and logs an Error message in the Windows System Event Log if the System hive approaches its maximum limit.

The next few sections explain the Service's capabilities and how to configure it for more advanced scenarios.

Installing the HP Registry Monitor Service

The Registry Monitor Service is shipped as part of a larger Management Agent Package (either SNMP Agents 6.2.1, or WBEM Providers 6.5). It comes in a single executable package that first uninstalls the previous versions of the agents and then installs the newer versions. In most situations, customization is not necessary. The advantage of this service is that it accurately determines the true size of the System hive as opposed to its indicated size on disk, and provides warnings should the size grow too large.

The HP Registry Monitor Service was added to HP Management Agents version 6.2.1. It is downloadable from the HP website as Smart Component `cp011353.exe`. It is supported on the following operating systems:

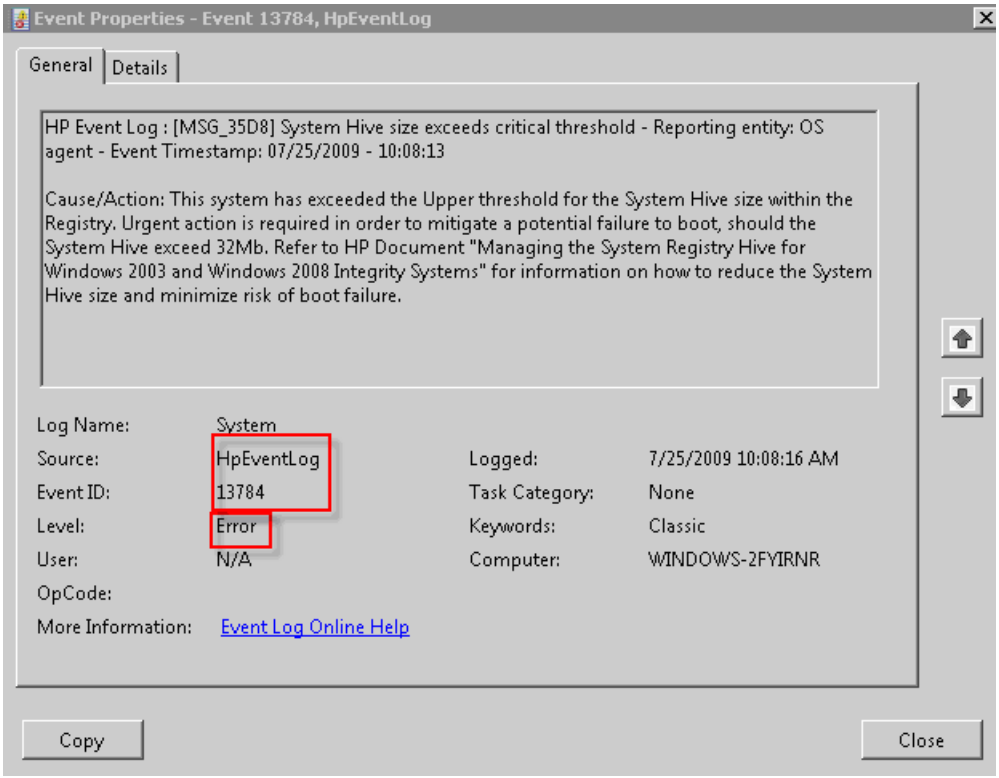
- Windows Server 2003 with Service Pack 2
- Windows Server 2008 with Service Pack 1 or Service Pack 2

The Registry Monitor Service is also part of the WBEM Providers package in the HP Integrity Support Pack version 6.5. Version 6.5 uses WBEM exclusively and is only supported on Windows Server 2008.

Configuring the HP Registry Monitor Service

The HP Registry Monitor Service, by default, checks the true size of the System hive and sends a Warning message to the Windows Event Log and to any configured Management Station when its size becomes greater than 24 MB. This warning is sent out once per day, typically at midnight, since that is when the file is checked. If the System hive continues to grow after the Warning message is sent, it replaces the Warning message with an Error message when the System hive exceeds 28 MB. Figure 12 shows the Error message it posts to the Windows System Event Log. Note that if the Registry Monitor Service was installed as part of SNMP Agents 6.2.1, the indicated source of the event will be "HpEventLog". If the Registry Monitor Service was installed as part of WBEM Providers 6.5, the indicated source of the event will be "HP System".

Figure 12 Error message



Default configuration of the thresholds, the time the System hive is checked, and the frequency of checking should be suitable for most configurations. However, if the System hive is checked and the file size is already over 24 MB, the service has to do some extra work to check the internal Hive size. As a result, the Registry is locked for a few seconds during this process. This behavior does not normally interfere with system operation since the System hive is checked by default at midnight, local time. However, if a critical job is already scheduled to run at that time, the service can be configured to run differently, as can the thresholds.

Configurable parameters for the service are stored in the following configuration/initialization file, which can be edited with `Notepad.exe`:

`c:\windows\system32\CPQMgmt\CqMgServ\hpmgtsvc.ini` (if the Registry Monitor Service was installed by SNMP Agents 6.2.1), or

`c:\Program Files\hpwbem\health\hpmgtsvcex.ini` (if the Registry Monitor Service was installed by WBEM Providers 6.5)

Notice the following section of the initialization file:

```
[REGMON
;NOTE: ONLY WHOLE NUMBERS ARE ACCEPTED IN THESE PARAMETERS
;This is the registry system hive size warning threshold in bytes, default 25165824
WARNING_THRESHOLD=25165824
;This is the registry system hive size critical threshold in bytes, default 29360128
CRITICAL_THRESHOLD=29360128
POLLTIME_SECONDS=86400
;This is the preferred local system time for registry size polling, in military (24h) format. Example 0100 for
 1am and 2000 for 8pm.
;Removing the semi-colon (;) before POLL_PREFERRED_TIME will enable it and will ignore POLLTIME_SECONDS above
POLL_PREFERRED_TIME=0000
QUIET_PERIOD_HOURS=24
```

Once the initialization file is edited and saved, the new values go into effect immediately without needing to restart the agents. After the Threshold values, perhaps the most useful value is POLL_PREFERRED_TIME. Change this value if you want the service to run at a more convenient time.

Deciding Whether to Implement a Single ControlSet strategy

Deploying a single ControlSet strategy (described in section, "System Recovery" (page 11)), can immediately reclaim approximately half the System hive space. You should deploy a single ControlSet strategy if the HP Registry Monitor Service has posted an Error message to the Windows System Event Log (similar to Figure 12), as the Hive is nearing its limit. Although this action will provide an immediate gain in space, it is still important to survey the System hive afterwards to determine which subkeys are the most costly, and see if they can be reduced. If the hive continues to grow without bound, it must be actively managed.

If a single ControlSet solution is implemented and the service continues to call out errors then it becomes mandatory to discover what is responsible for the growth of the hive. If it is impossible to circumvent this growth through active management, the next step must be to deploy Windows Server 2008 R2 Itanium Edition, which supports much greater hive sizes.

Optimizing the Hive

Moving to a single ControlSet solution reduces the hive size greatly, yet its disk size remains the same. This is not optimal for the HP Registry Monitor Service because it has to run a more in-depth analysis of the hive to determine its true size. It is possible, once the redundant ControlSet(s) are removed, to compress the System hive down, but this requires a reboot into WinPE.

In order to reduce the hive's disk size, you must open RegEdit and highlight the System key. Then export the System hive to a compressed version by using the **Export Hive** option as shown in Figure 3 (page 9), making sure to save the file as a hive file.

The System can now be rebooted into WinPE, and the saved hive file can be saved as the new System hive using the following commands:

```
rename c:\windows\system32\config\system c:\windows\system32\config\
system.big
```

```
c:\windows\system32\config\system.tiny c:\windows\system32\config\system
```

If the exported System hive is less than 24 MB then the HP Registry Monitor Service does not need to do an in-depth analysis of the file.

Conclusion

The System hive limit of 32 MB is generally avoidable with care, knowledge, and management of the contents of the hive ControlSet(s). Specifically, it is imperative to know the size of the relevant disk keys and whether they have the correct number of entries. With this knowledge, the Hive can be managed relatively easily, and if configured for a single ControlSet with the correct number of paths, the system can support very large numbers of disks and partitions.