Virtual Machine Generation ID

August 1, 2012

Abstract

This paper provides information about virtual machine generation ID capability for Windows Server 2012 or Windows 8. It also provides guidelines for developers to implement this capability in virtualization platforms.

This information applies to the following operating systems:
 Windows Server® 2012

 Windows® 8

The current version of this paper is maintained on the web at:
 <http://go.microsoft.com/fwlink/?LinkId=260709>

**Disclaimer**: This document is provided “as-is”. Information and views expressed in this document, including URL and other Internet website references, may change without notice. Some information relates to pre-released product which may be substantially modified before it’s commercially released. Microsoft makes no warranties, express or implied, with respect to the information provided here. You bear the risk of using it.

Some examples depicted herein are provided for illustration only and are fictitious. No real association or connection is intended or should be inferred.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

© 2012 Microsoft. All rights reserved.



Contents

[Introduction 3](#_Toc332041690)

[Why is a virtual machine generation ID needed? 3](#_Toc332041691)

[Cryptographic software 3](#_Toc332041692)

[Vector-clock synchronization software 3](#_Toc332041693)

[How is the virtual machine generation ID used in Windows Server 2012 or Windows 8? 4](#_Toc332041694)

[Windows Cryptography API: Next Generation 4](#_Toc332041695)

[Active Directory 4](#_Toc332041696)

[How to implement virtual machine generation ID support in a virtualization platform 4](#_Toc332041697)

# Introduction

This document discusses a new capability in Windows Server 2012 or Windows 8 that is call the virtual machine generation ID. This functionality is new to Hyper-V in Windows Server 2012 or Windows 8 and is designed to address some key problems that arise from virtual machine operations that have the effect of returning a virtual machine to an earlier point in time (like applying a virtual machine snapshot).

The virtual machine generation ID is a feature whereby the virtual machines BIOS will expose a new ID. This is a 128-bit, cryptographically random integer value identifier that will be different every time the virtual machine executes from a different configuration file—such as executing from a recovered snapshot, or executing after restoring from backup. Programs running in a virtual machine can protect themselves from potential corruption by checking that the generation ID has not changed immediately prior to committing a transaction, they can also use the data provided in the 128-bit identifier as a high entropy random data source.

# Why is a virtual machine generation ID needed?

There are two classes of software that can be negatively affected by virtual machine operations that have the effect of returning a virtual machine to an earlier point in time (like applying a virtual machine snapshot).

## Cryptographic software

Windows provides APIs that offer random data with a high level of entropy for use by a range of software applications. The logic for creating this random data is captured, and restored, in certain virtual machine operations (like taking and applying an online snapshot). This has the effect of severely reducing the entropy of the random data provided by Windows.

A practical example of this is globally unique identifier (GUID) generation.

Many applications rely on Windows to provide unique GUID values when requested. However, if a GUID is immediately requested after repeated applications of a virtual machine snapshot, there is a high probability of the application receiving a duplicate GUID value.

## Vector-clock synchronization software

Recovering a virtual machine from a snapshot can corrupt applications that use vector-clock synchronization. Example products include Active Directory, DFS-R, SQL replication, and the sync framework. For these protocols to work, each member of the replica set must maintain a monotonically increasing logical clock. When a user restarts a virtual machine snapshot, it rolls back the logical clock of the replica member in the virtual machine, causing clock values to get associated to different transactions. As a consequence, members of the replica set will not converge to the same state. To the administrator or end-user, this is data corruption.

# How is the virtual machine generation ID used in Windows Server 2012 or Windows 8?

Virtual machine generation ID functionality is available to any software running inside of a virtual machine. In Windows Server 2012 or Windows 8, there are two components that are using this new capability.

## Windows Cryptography API: Next Generation

The Windows Cryptography API: Next Generation ([http://msdn.microsoft.com/en-us/library/windows/desktop/aa376210(v=vs.85).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/aa376210%28v%3Dvs.85%29.aspx)) in Windows Server 2012 or Windows 8 use the data provided from the virtual machine generation ID to ensure that applications that use these APIs always receive high entropy for key and random number generation, even in the event of a virtual machine snapshot or similar operation.

## Active Directory

Virtual environments present unique challenges to distributed workloads that depend upon a logical clock-based replication scheme. Active Directory Directory Services replication, for example, uses a monotonically increasing value (known as a USN or Update Sequence Number) assigned to transactions on each domain controller. Each domain controller's database instance is also given an identity, known as an InvocationID. The InvocationID of a domain controller and its USN together serve as a unique identifier associated with every write-transaction performed on each domain controller and must be unique within the forest.

Active Directory Directory Services replication uses InvocationID and USNs on each domain controller to determine what changes need to be replicated to other domain controllers. If a domain controller is rolled back in time outside of the domain controller's awareness and a USN is reused for an entirely different transaction, replication will not converge because other domain controllers will be out of sync with those changes. A virtual machine makes it easy for hypervisor administrators to roll back a domain controller's USNs (its logical clock) by, for example, applying a snapshot outside of the domain controller's awareness.

Beginning with Windows Server 2012, Active Directory Directory Services virtual domain controllers hosted on hypervisor platforms that support virtual machine generation ID functionality can detect and employ necessary safety measures to protect the Active Directory Directory Services environment if the virtual machine is rolled back in time by the application of a virtual machine snapshot, or similar operation.

You can read more about this here: <http://technet.microsoft.com/library/hh831734.aspx>

# How to implement virtual machine generation ID support in a virtualization platform

Any hypervisor can and should project this feature on a virtual machine to expose the desired functionality in Windows. Specifically, the BIOS must:

1. Put the generation ID in an 8-byte aligned buffer in guest RAM, ROM, or device memory space, which is guaranteed not to be used by the operating system. It must not be in ranges reported as AddressRangeMemory or AddressRangeACPI. It must not be in the same 4-kilobyte page as any memory that is expected to be mapped by a page table entry with caching disabled.

2. Expose a device somewhere in the ACPI namespace with a compatible ID (\_CID) of "VM\_Gen\_Counter".

3. Include a "DOS Device Name" ID (\_DDN) of "VM\_Gen\_Counter".

4. Include an object below this device named "ADDR", which evaluates to a package of two integers. The first integer must be the low 32-bits of the guest physical address of the buffer that contains the 128-bit generation ID. The second integer must be the high 32-bits of the guest physical address of the buffer that contains the generation ID.

5. When the generation ID changes, execute an ACPI Notify operation on the generation ID device (for example, by triggering an ACPI GPE), using notification code 0x80. Note that the BIOS may issue the notify operation even if the generation ID has not changed.

A hypervisor must also expose an object below the device, named "\_HID", which is the "hardware ID." This value is hypervisor specific and must be different for each vendor. The Windows generation ID guest driver will load on the compatible ID of "VM\_Gen\_Counter", so it will be present regardless of which hypervisor it is running on top of, as long as that hypervisor follows the rules above.

After the identifier has been made persistent in the configuration and exposed in the guest, we can see how operations on the virtual machine can cause workloads in a guest to observe a difference between the generation ID of the current execution context and one that had been previously stored with the workload dataset.

**Application of online snapshots** – Applying an online snapshot will cause a previous workload time interval to be re-executed. The workload can opportunistically, but not deterministically, detect a generation change with respect to any attached data set.

**Application of offline snapshots** – Applying an offline snapshot will cause a previous workload time interval to be re-executed. The workload can deterministically detect the generation change with respect to any attached data set by sampling the value at workload startup, before any workload transactions are processed.

**Restoring from backup** – Restoring a backup image will cause a previous workload time interval to be re-executed. Upon restore, the components of the backup are enumerated and replaced on the restore target by the VSS system. The affected configuration files are simply copied and not re-realized on the host. The restore sequence will be modified to post process the restore target and apply new generation identifiers to the restored configuration files.

**Failover to replication target** – A failover to a replication target could cause a previous workload time interval to be re-executed. When a failover target is established, the configuration of the source virtual machine is transmitted to the target and realized, resulting in the assignment of a new generation identifier. If the workload is ever resumed on the target, the generation ID will properly denote the time shift.

**Importing a virtual machine** – Whether an imported virtual machine represents the most current execution context cannot be determined by the virtual machine host. It is assumed that future execution could have occurred and will generate new generation identifiers for all imported configurations.

|  |  |
| --- | --- |
| Scenario | Generation ID changed |
| Virtual machine is paused or resumed | No |
| Virtual machine reboots | No |
| Virtual machine host reboots | No |
| Virtual machine starts executing a snapshot (every time) | Yes |
| Virtual machine is recovered from backup | Yes |
| Virtual machine is failed over in a disaster recovery environment | Yes |
| Virtual machine is live migrated | No |
| Virtual machine is imported, copied, or cloned | Yes |
| Virtual machine is failed over in a clustered environment | No |
| Virtual machine's configuration changes | Unspecified[[1]](#footnote-1) |

1. The virtualization platform may change the generation ID if it determines that software in the virtual machine needs to be alerted to the potential for problems. [↑](#footnote-ref-1)