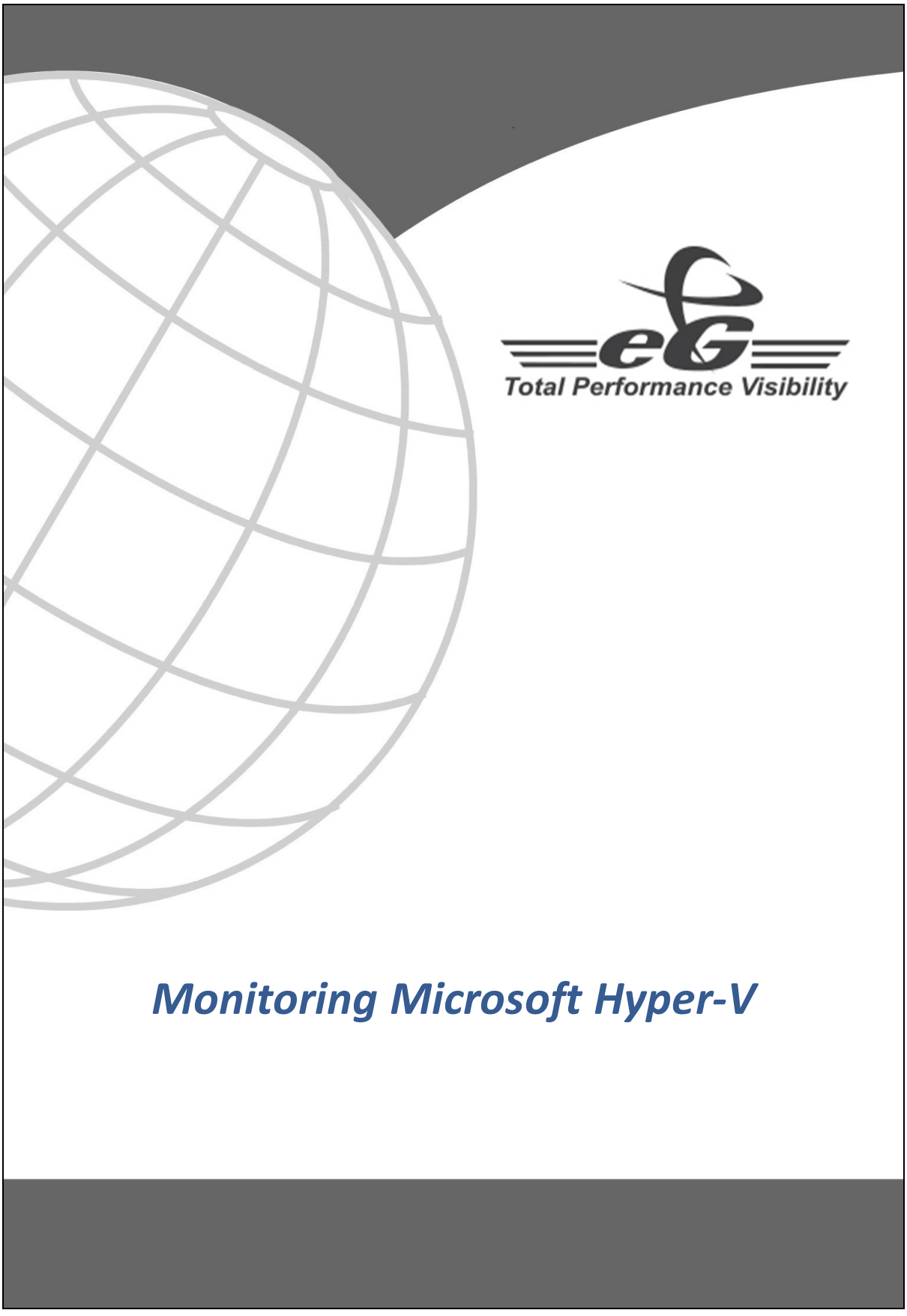


Monitoring Microsoft Hyper-V

Table of contents

Table of Figures

Table of Contents

1.1 How eG Enterprise Monitors Microsoft Hyper-V Servers?

eG Enterprise offers two specialized monitoring models – one each for each of the distinct deployment architectures of Hyper – V. While the generic *Hyper-V* model is to be used for monitoring Hyper-V servers with VMs hosting server applications, the *Hyper-V VDI* model is ideal for virtual desktop environments.

Regardless of the model being used, eG Enterprise adopts a patented *In-N-Out* approach to monitoring it. This approach enables administrators to monitor the Hyper-V server inside out and determine the following:

- The overall health of the Hyper-V host
- The physical resource usage by the Hyper-V host and host processes
- Whether critical Hyper-V services are available or not;
- The availability of the Hyper-V server
- The current status of the VMs configured on the host and the the fraction of physical resources used up by each VM, as seen from outside the VMs; this represents the “outside” view
- The fraction of allocated resources used up by each VM; this represents the “inside” view

1.2 Agent Deployment Model

Using eG Enterprise, administrators can manage a Hyper-V server in an agent-based manner.

The agent-based approach requires that an eG agent be installed on the root partition of the Hyper-V server. The root partition runs a Windows 2008 64-bit operating system. Therefore, to monitor the Hyper-V server, you need to install the Windows 2008 64-bit agent on the root partition. The steps for the installation are clearly laid out in the eG Installation Guide document. The agent then uses Perfmon to extract metrics from the Hyper-V host, auto-discovers the IP addresses of the guests on the host, communicates with every guest via WMI, and then collects the “inside view” metrics using Perfmon (see Figure 1.1).

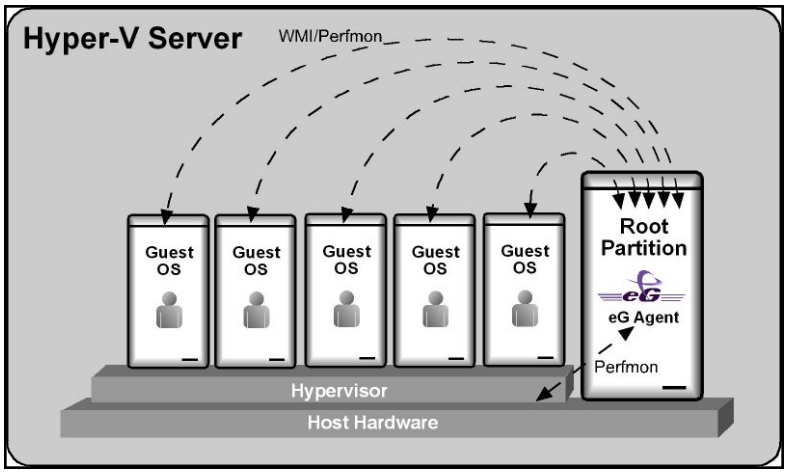


Figure 1.1: Agent-based monitoring of Hyper-V

Note:

The eG agent can collect "inside view" metrics from Windows VMs alone; for Linux VMs, only the powered-on status and "outside view" metrics will be available.

The eG agent then communicates remotely with every Windows VM on the target Hyper-V host (using WMI) to obtain the inside view of the VMs. To establish this remote connection with Windows VMs, eG Enterprise requires that the eG agent be configured with domain administrator privileges. In high-security environments, where the IT staff might have reservations about exposing the credentials of their domain administrators, this approach to extracting "inside view" metrics might not be preferred. In such environments therefore, eG Enterprise provides administrators the option to deploy a piece of software called the **eG VM Agent** on every Windows VM, which allows the eG agent to collect "inside view" metrics from the VMs **without domain administrator rights.** Refer to Section 1.3.2 for more details on the **eG VM Agent**.

For a detailed list of pre-requisites for monitoring Hyper-V, refer to Section 1.3.

1.3 Pre-requisites for Monitoring Microsoft Hyper-V

There are several pre-requisites for an eG agent to be able to monitor a Hyper-V server and the guest VMs hosted on it.

- The eG agent on the root partition should be able to communicate with the eG manager port (default is 7077).
- The **Integration Services** component should be installed on every VM to be monitored, so that the IP address of the VMs is discovered; also, the IP address should be resolvable in DNS.

  Note:

  - The eG agent can automatically discover the IP address of the Windows VMs only; for Linux VMs, only the name of the VM will be discovered and not its IP address.
  - If multiple IP addresses are configured on a single Windows VM, the eG agent will discover only one of the IP addresses and not all of them.

- The **Integration Services** component provides a **Key/Value Pair Exchange** script, which the eG agent uses for discovering the IP addresses of the VMs on the Hyper-V host. Since this script is supported only on specific Windows versions, the eG agent can discover the IP address of the VMs executing on those versions only. The supported Windows versions are as follows:
    - Windows Server 2008 64-bit
    - Windows Server 2008 x86
    - Windows Server 2003 x64 with SP2
    - Windows 2000 Server with SP4
    - Windows 2000 Advanced Server SP4
    - Windows Vista x64 with SP1
    - Windows Vista x86 with SP1
    - Windows XP x86 with SP2/SP3
    - Windows XP x64 with SP2

➢ To obtain the "inside view" of Windows VMs **without using the 'eG VM Agent'**, the following will have to be performed:

- The **ADMIN$** share should be enabled for all Windows virtual guests being monitored and the administrative account must have permissions to this share drive. Refer to Section for a step-by-step procedure to achieve this.
- All "inside view" tests run by the eG agent should be configured with the credentials of a domain administrator.
- Set the **INSIDE VIEW USING** flag for all the "inside view" tests to **Remote connection to VM (Windows)**.

➢ To obtain the "inside view" of Windows VMs **using the 'eG VM Agent'**, follow the steps given below:

- Install the eG VM Agent on every Windows VM to be monitored; the procedure for installing the eG VM Agent are detailed in Section 1.3.2 of this document.
- Enable the eG agent to communicate with the port at which the eG VM Agent listens (the default port is 60001).
- Set the **INSIDE VIEW USING** flag for all the "inside view" tests to **eG VM Agent (Windows)**.

1.3.1 Configuring Windows Virtual Machines to Support the eG Agent's Inside View without the eG VM Agent

For the "inside" view, by default, the eG agent communicates remotely with the virtual machines on the Hyper-V server and collects metrics. To establish this remote connection with Windows VMs, eG Enterprise requires that the eG remote agent (on Windows) be configured with domain administrator privileges. Besides, the inside view using flag of all "inside view" tests should be set to **Remote connection to a VM (Windows)**.

In addition, the **ADMIN$** share will have to be available on the Windows guests.

If the **ADMIN$** share is not available on any Windows-based virtual guest, create the share using the procedure detailed below:

1. Open the Windows Explorer on the virtual machine, browse for the corresponding **Windows** directory in the C drive, right-click on it, and select the **Sharing** option from the shortcut menu.
2. If the **ADMIN$** share does not pre-exist on the Windows guest, then 1.3.1 appears indicating the same.

Figure 1.2: The ADMIN$ share does not exist

3. On the other hand, if the **ADMIN$** share pre-exists, Figure 1.3 appears. In such a case, first, remove the **ADMIN$** share by selecting the **Do not share this folder** option from Figure 1.3 and clicking the **Apply** and **OK** buttons. After this, you will have to repeat step 1 of this procedure to open 1.3.1. Then, proceed as indicated by step 3 onwards.

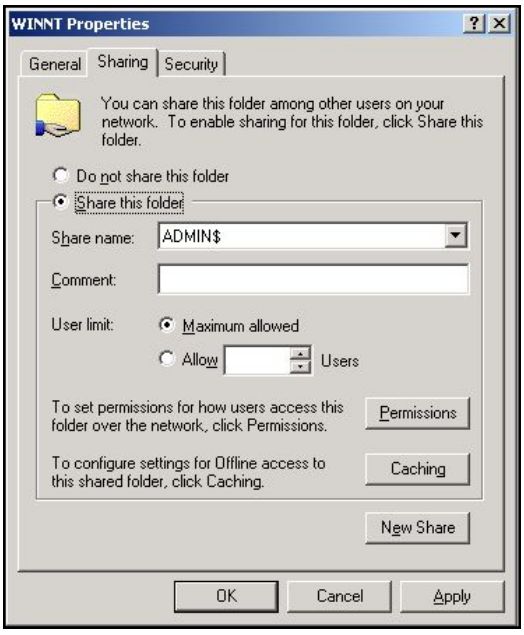


Figure 1.3: Admin$ share pre-exists

4. To create (or re-create) the **ADMIN$** share, select the **Share this folder** option from Figure 1.3, and provide **ADMIN$** share against the **Share name** text box (see Figure 1.4).

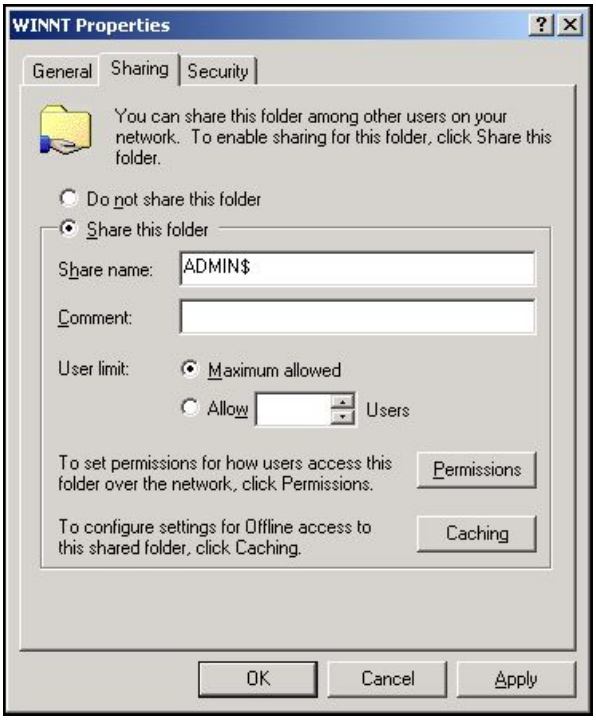


Figure 1.4: Creating the ADMIN$ share

5. Next, to enable the eG agent to communicate effectively with the Windows guest, you need to ensure that the permission to access the **ADMIN$** share is granted to an administrative user (domain); also, the

credentials of this user should be passed while configuring the eG monitoring capabilities - i.e., while configuring the Hyper-V tests. To grant the access permissions, click on the **Permissions** button in Figure 1.4.

6. By default, the **ADMIN$** share can be accessed by **Everyone** (see Figure 1.5). To grant access rights to a specific domain administratoar, select the **Add** button in Figure 1.5. When Figure 1.6 appears, select the domain to search from the **Look in** list. The valid user accounts configured on the chosen domain then appear in the box below. From this box, choose the administrator's account and click on the **Add** button to add the chosen user account to the box below the **Add** button.

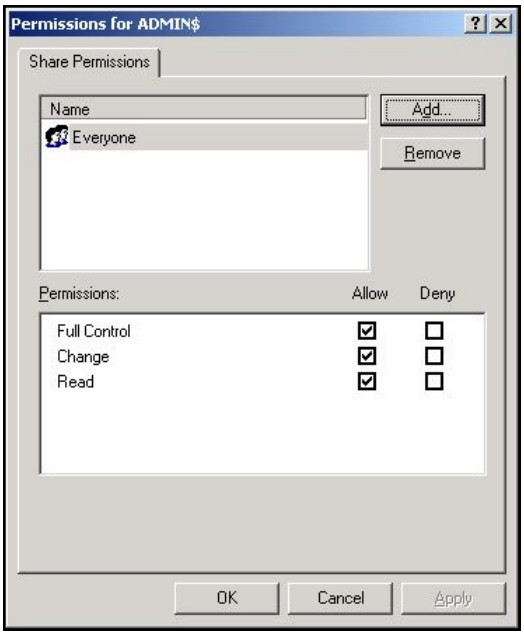


Figure 1.5: Clicking the Add button

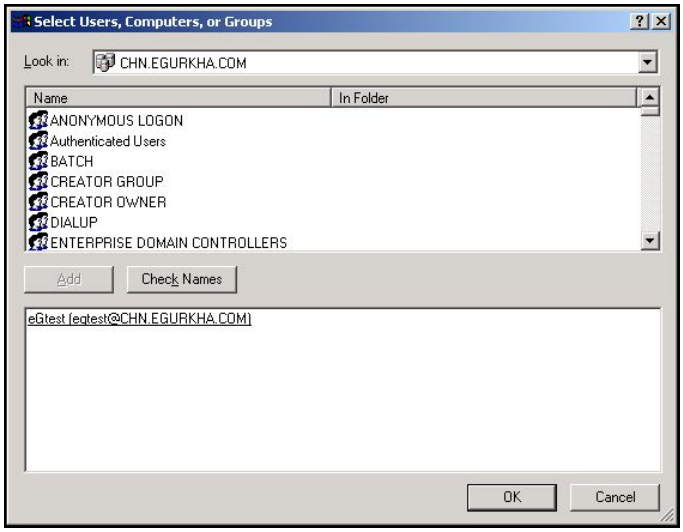


Figure 1.6: Selecting the administrative user to whom access rights are to be granted

7. Finally, click the **OK** button. You will then return to Figure 1.5, where the newly added administrator account will appear (see Figure 1.7).

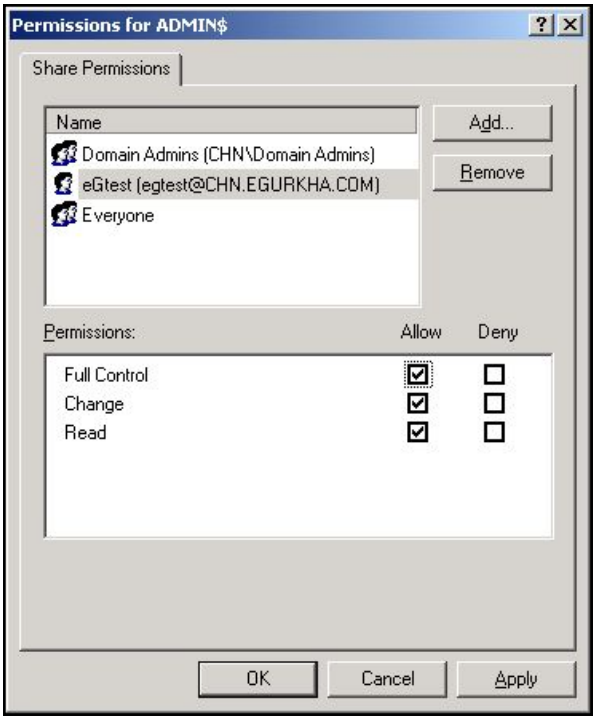


Figure 1.7: The administrator account granted access permissions

8. Select the newly added administrator account from Figure 1.7, and then, using the **Permissions** section, grant the administrator **Full Control**, **Change**, and **Read** permissions.
9. Finally, click the **Apply** and **OK** buttons in Figure 1.7 to register the changes.
10. Once you return to Figure 1.7, click on the **Security** tab (see Figure 1.8) to define the security settings for the **ADMIN$** share.

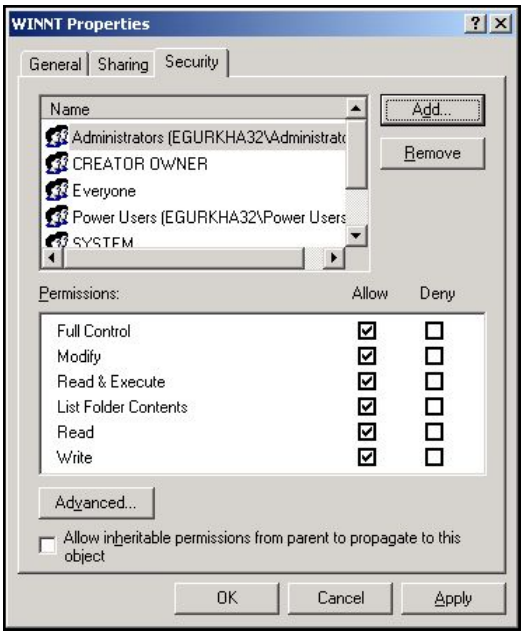


Figure 1.8: Defining the Security settings for the ADMIN$ share

11. Here again, you need to add the same administrator account, which was granted access permissions earlier. To do so, click the **Add** button in Figure 1.8, pick a domain from the **Look in** list of Figure 1.9, select the said administrator account from the domain users list below, and click the **Add** button (in Figure 1.9) to add the chosen account. Then, click the **OK** button in Figure 1.9.

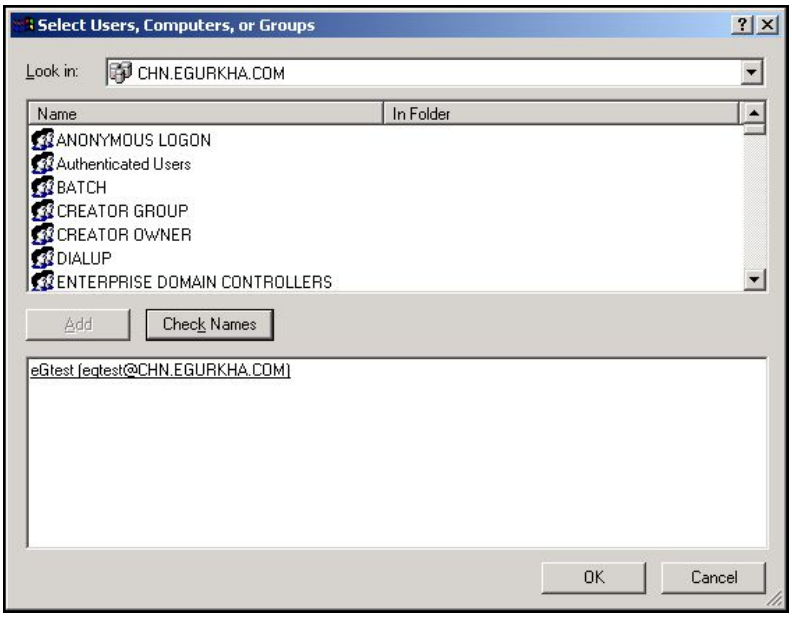


Figure 1.9: Adding the administrator account

12. This will bring you back to Figure 1.8, but this time, the newly added domain administrator account will be listed therein as indicated by Figure 1.10.

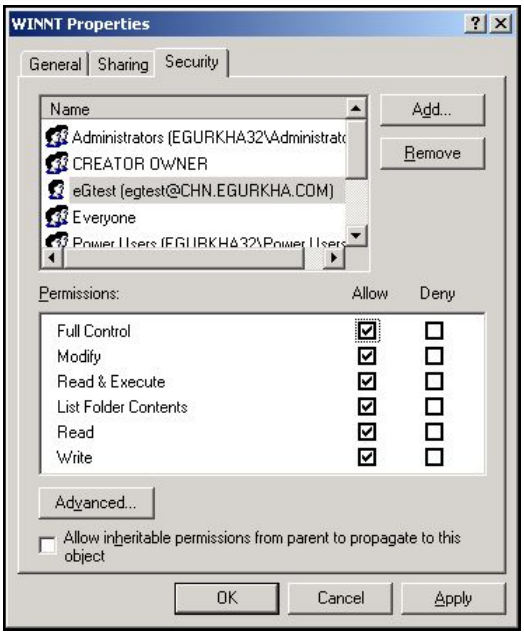


Figure 1.10: The Administrator account in the Security list

13. Finally, click the **Apply** and **OK** buttons in Figure 1.10.

1.3.2 Configuring Windows Virtual Machines to Support the eG Agent's Inside View Using the eG VM Agent

To obtain the inside view of a Windows VM, the eG agent offers two options. The first option uses Windows File & Print Sharing services to push monitoring components to the VMs. These monitoring components are then executed on the VM to collect metrics from the VMs. To push monitoring components to the VM and to periodically invoke these components, the eG agent requires **domain administrator privileges** to all the VMs being monitored.

In many production environments, strict security restrictions are enforced, and it may not be possible to configure a monitoring solution with domain administration privileges for each of the VMs. To handle such environments, the eG VM monitor uses a lightweight monitoring component called the **eG VM Agent**, which is installed inside each of the VMs to obtain metrics regarding the health of the VMs. The eG VM Agent can be best described as a software that can be installed on the Windows virtual machines of a virtual infrastructure to allow a single eG agent to obtain an inside view of these VMs, without domain administrator privileges.

1.3.3 Installing the eG VM Agent

Users have multiple options to choose from when it comes to installing the eG VM Agent. These options have been discussed below:

- Manually install the eG VM Agent on every Windows VM using the executable that eG Enterprise provides;

- Bundle the eG VM Agent as part of a template VM, and use this template to create multiple VMs; this way, the eG VM Agent is automatically available in all the VMs that are created using the template;
- Use a software distribution solution such as Microsoft System Center to distribute the eG VM Agent software to existing VMs from a central location;
- Connect to each Windows VM and silently install the eG VM Agent on it, without using the executable that eG Enterprise provides.

The first and fourth installation options alone are discussed here.

1.3.3.1 Using the Executable Provided by eG Enterprise

The detailed manual installation procedure has been discussed hereunder:

1. To install the eG VM Agent on a 32-bit VM, double-click on the **eGVMAgent.exe**, and to install the same on a 64-bit VM, double-click the **eGVMAgent_64.exe**.
2. Figure 1.11 then appears. Click on the **Next** button in Figure 1.11 to continue.

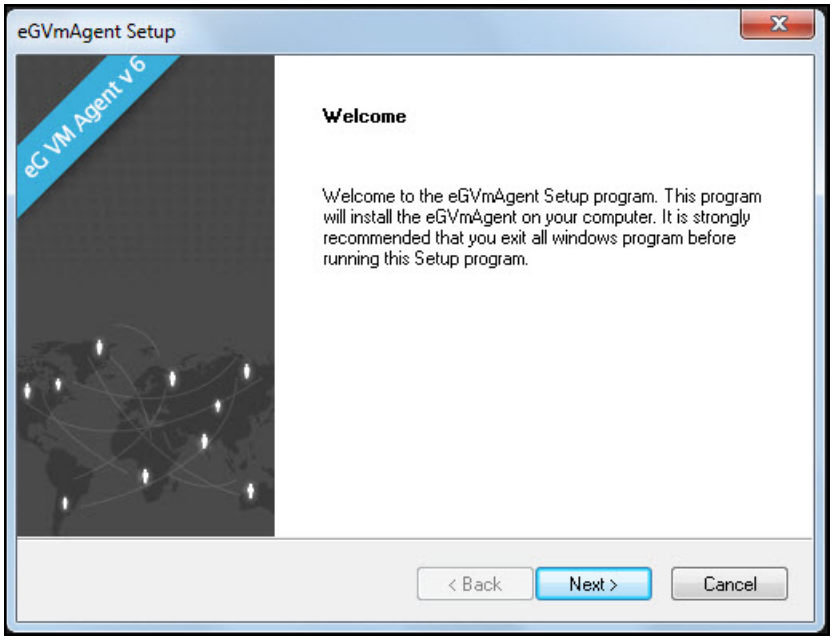


Figure 1.11: Welcome screen of the eG VM Agent installation wizard

3. When Figure 1.12 appears, click on **Yes** to accept the displayed license agreement.

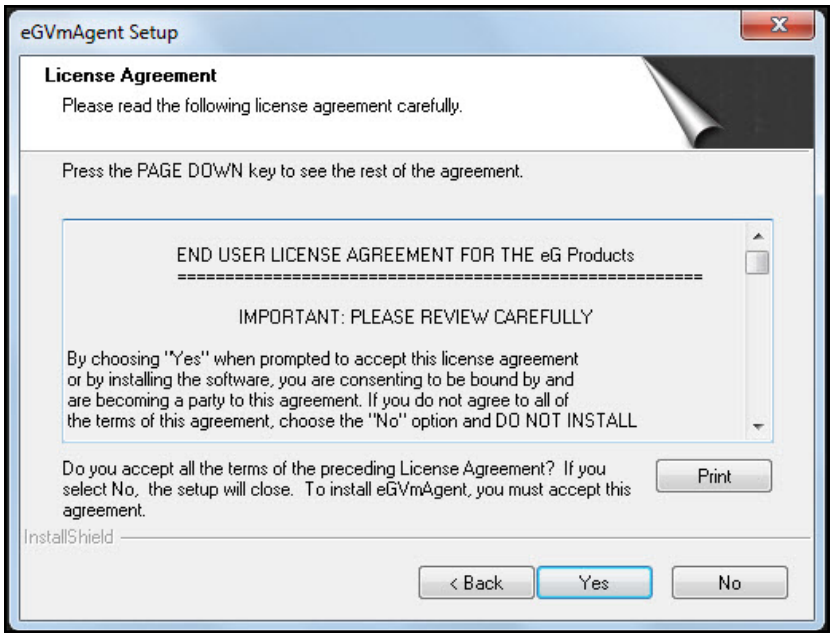


Figure 1.12: Accepting the license agreement

4. Use the **Browse** button in Figure 1.13 to indicate the location in which the agent should be installed, and click the **Next** button to proceed.

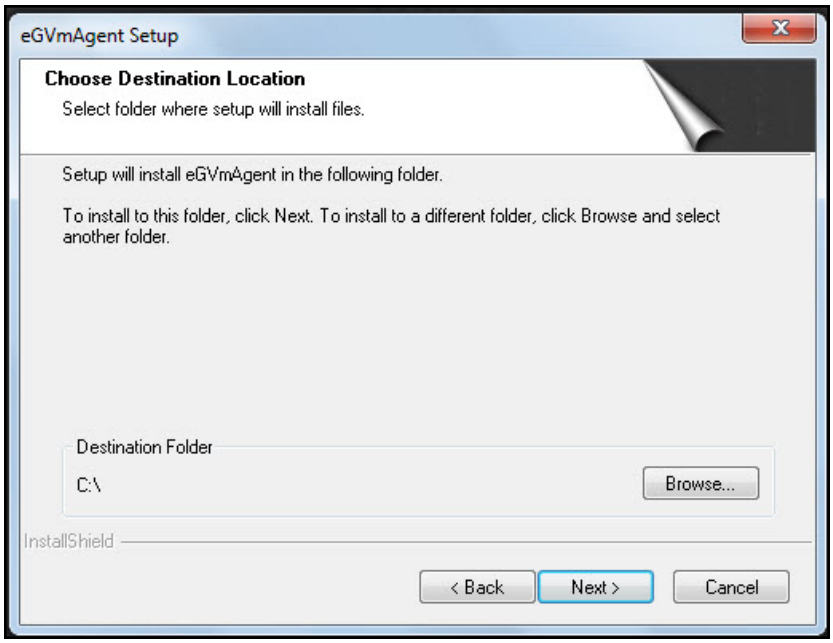


Figure 1.13: Specifying the install directory of the eG VM Agent

5. Next, specify the port at which the VM agent listens for requests from the eG agent. The default port is 60001. After port specification, click on the **Next** button in Figure 1.14 to proceed.

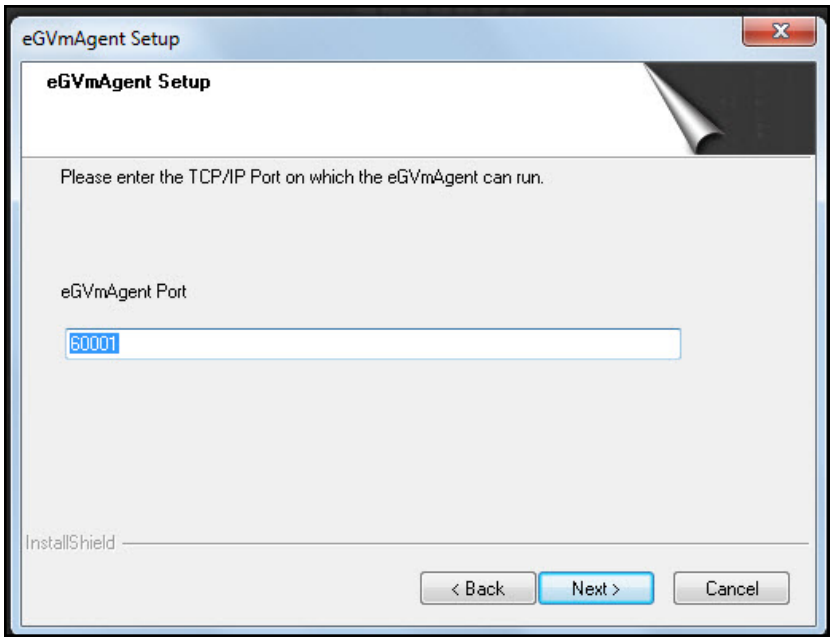


Figure 1.14: Specifying the VM agent port

6. A summary of your specifications then follows (see Figure 1.15). Click **Next** to proceed.

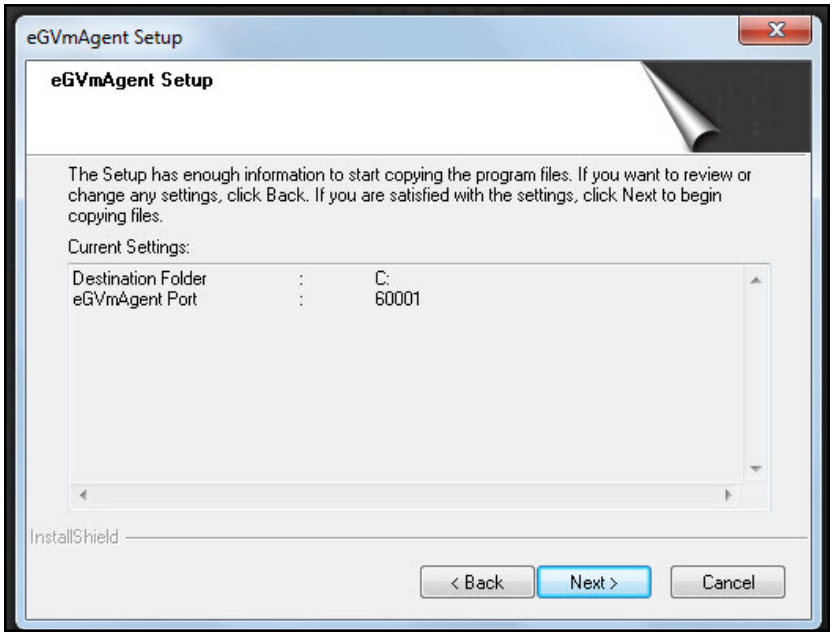


Figure 1.15: A summary of your specifications

7. Finally, click the **Finish** button in Figure 1.16 to complete the installation.

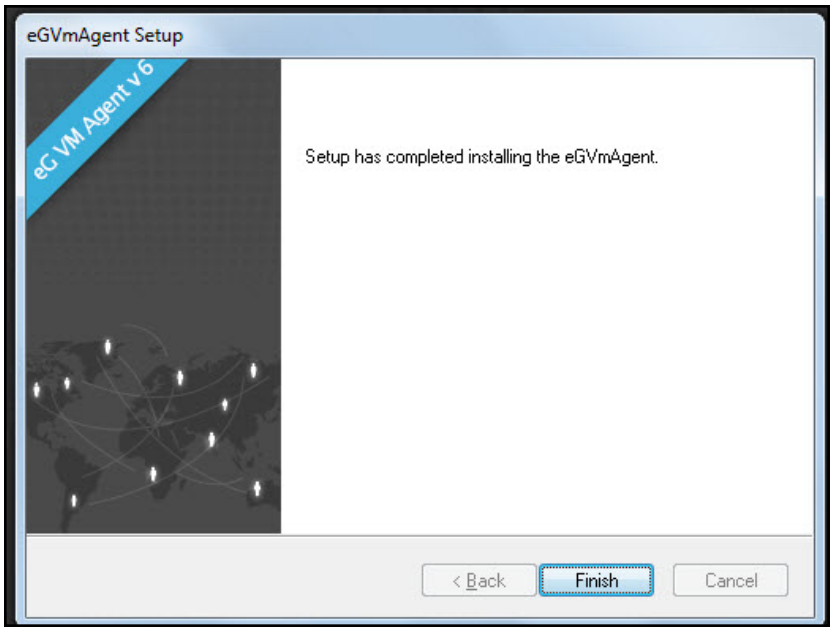


Figure 1.16: Finishing the installation

1.3.4 Silent Installation of the eG VM Agent

To silently install the eG VM agent on Windows VMs, follow the broad steps outlined below:

1. Creating silent mode script for eGVmagent installation
2. Installing eGVmAGent in silent mode

Each of these steps have been discussed elaborately below.

1.3.4.0.1 Creating a Silent Mode Script

For this, follow the procedure detailed below:

1. Login to a target Windows VM.
2. From the command prompt, run the following command to launch the normal mode installation of the eG VM Agent.

   *eGVMAgent_**<32/64>**.exe /a /r /f1**"<Full path to the script file into which the installation inputs will be stored>"***

   For example:

   eGVMAgent_x64.exe /a /r /f1"C:\script\eGVMAgent.iss"
3. Upon execution, this command will automatically create a script file of the given name in the location mentioned in the command.
4. Command execution will also begin the normal mode installation of the eG VM Agent. Provide inputs as

and when necessary to proceed with the installation.

5. These inputs will be automatically recorded in the script file that was created in step 3.

1.3.4.0.2 Installing the eG VM Agent in the Silent Mode

Follow the steps given below to install the eG VM Agent in the silent mode:

1. Login to the Windows VM where the script file containing the inputs for installation resides.
2. Copy the script file from this VM to the Windows VM on which you want to install the eG VM Agent in the silent mode.
3. Copy the eG VM Agent installation executable also to the target Windows VM.
4. Next, on the target Windows VM, run the following command from the command prompt:

   *eGVMAgent_* ***<32/64>****.exe /a /s /f1"****<Full path to the script file containing the inputs for the installation>****"*

   For example:

   eGVmAgent_x64.exe /a /s /f1"C:\script\eGVMAgent.iss"

5. Upon successful execution, this command will automatically install the eG VM Agent on the target Windows VM.
6. You can then repeat steps 1-5 on each Windows VM where you want to install the eG VM Agent.

1.3.4.1 Communication between the eG Agent and the eG VM Agent

At the time of the installation of the eG VM agent, a folder named eGVMAgent is created in the install destination specified. The setup program also creates a Windows Service named eGVMAgent on the Windows VM. This service must be running for the eG agent to obtain the inside view of the virtual machine.

Upon successful installation, the eG VM agent starts automatically and begins listening for requests at default TCP port 60001. However, if, during the installation process, you have configured a different port for the eG VM agent, then, after completing the installation, follow the steps below to make sure that the eG agent communicates with the eG VM agent via the port that you have configured:

- Login to the eG manager host.
- Edit the **eg_tests.ini** file in the **<EG_INSTALL_DIR>\manager\config directory**.

- The **WmiInsideViewPort** parameter in the **[AGENT_SETTINGS]** section of the file is set to **60001** by default. If the eG VM agent's port is changed at the time of installation, then you will have to ensure that this parameter reflects the new port. Therefore, change the default port specification accordingly.
- Save the file.

At configured intervals, the eG remote agent issues commands to each of the eG VM Agents (using the TCP port configured during the VM agent installation). The eG VM Agent executes the commands, collects the "inside view" metrics from the Windows VM, and sends the output back to the eG agent. The eG agent then analyzes the metrics and informs the eG manager about the status of the Windows VMs.

1.3.5 Licensing of the eG VM Agent

The eG VM Agent is not license-controlled. Therefore, you can install and use any number of VM agents in your infrastructure.

1.3.5.1 Benefits of the eG VM Agent

The eG VM Agent offers several key benefits:

- Ideal for high-security environments: The eG VM Agent is capable of collecting "inside view" metrics from Windows VMs, without domain administrator privileges. It is hence ideal for high-security environments, where administrators might not be willing to expose the credentials of the domain administrators.
- Easy to install, configure: The eG VM Monitor offers users the flexibility to choose from multiple methodologies for installing the eG VM Agent on the target VMs. Even a manual installation procedure, would not take more than a few minutes. Moreover, since the eG VM agent communicates only with the eG agent and not the eG manager, no additional configuration needs to be performed on the VM agent to facilitate the communication. In addition, the VM agent starts automatically upon installation, thereby saving the time and trouble involved in manually starting each of the VM agents.
- License independent**:** Since the eG VM agent is not license-controlled, you can add any number of VM agents, as and when required, to your environment.

The chapters to come elaborately discuss the *Microsoft Hyper-V* and the *Microsoft Hyper-V VDI* models that eG Enterprise offers.

Chapter

2

2

Administering eG Enterprise to Monitor a Microsoft Hyper-V Server

To monitor a Microsoft Hyper-V / Hyper-V VDI server using eG, follow the steps below:

1. Log into the eG administrative interface.
2. The eG manager is capable of auto-discovering the Hyper–V / Hyper-V VDI server. If these servers are already discovered, then use the Infrastructure -> Components ->Manage/Unmanage menu to manage them. Otherwise run discovery process using the menu sequence: Infrastructure -> Components -> Discover and manage the Hyper-V servers as detailed in Figure 2.1 to Figure 2.4. While Figure 2.1 and Figure 2.2 depict how to manage an auto-discovered Hyper-V server, Figure 1.3 and Figure 1.4 show how to manage an Hyper-V VDI server using eG. As shown, first select the **Component type**, then pick the servers to be managed from the **Unmanaged Components** list, and click the **<** button to manage the servers. Finally, click the **Update** button.

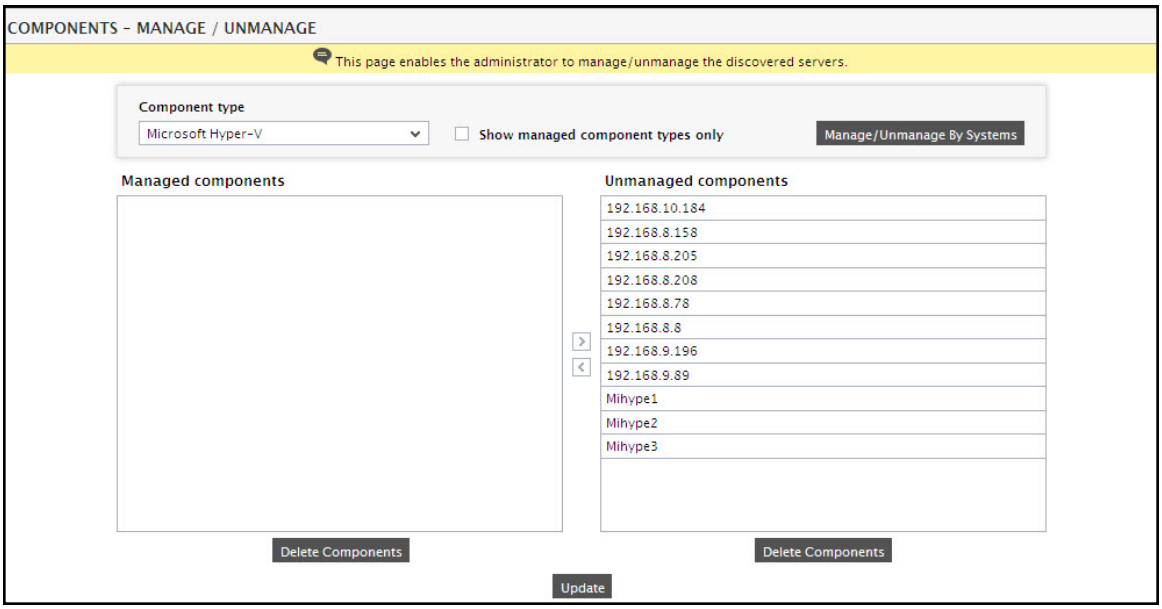


Figure 2.1: Viewing the unmanaged Hyper-V Servers

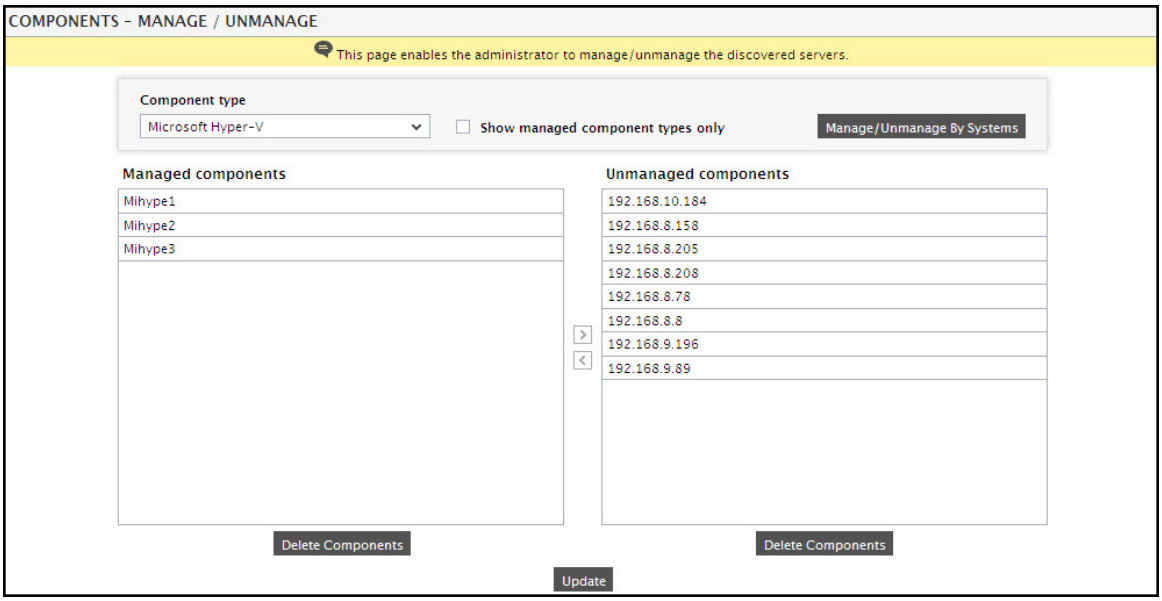


Figure 2.2: Managing the Hyper-V Servers

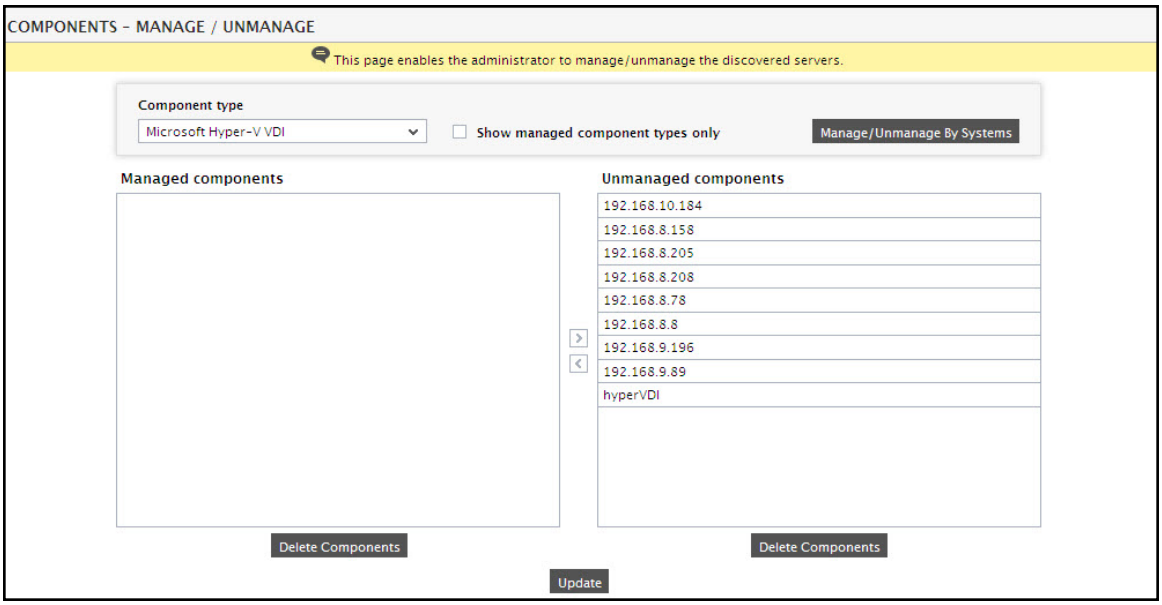


Figure 2.3: Viewing the unmanaged Hyper-V VDI Servers

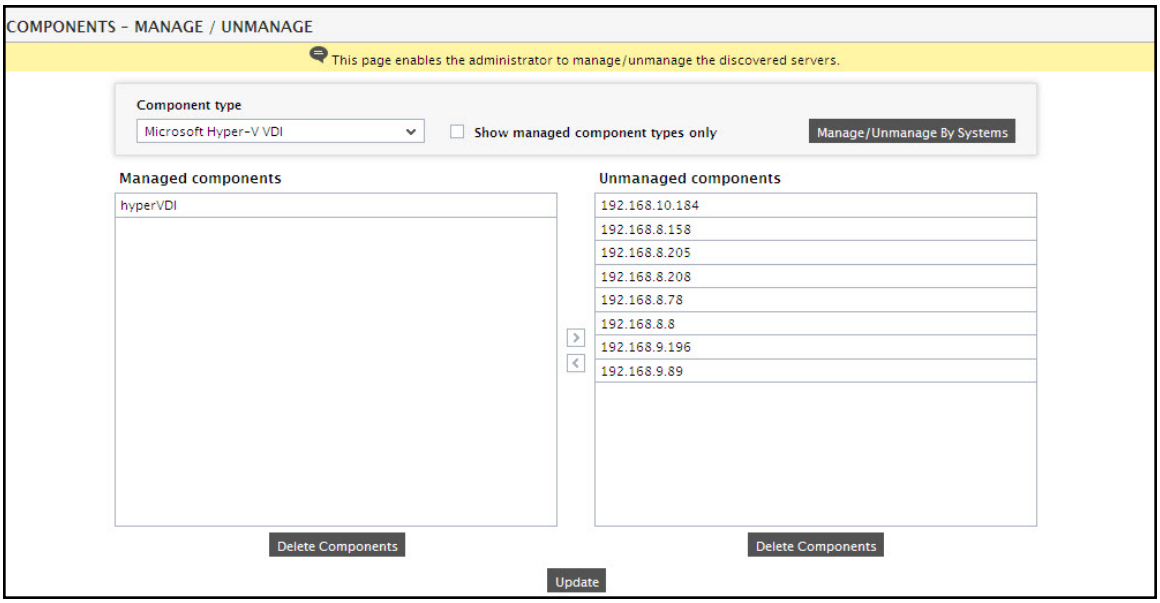


Figure 2.4: Managing the Hyper-V VDI Servers

3. Alternatively, you can also manually add the target Hyper-V / Hyper- VDI servers using the **COMPONENTS** page (see Figure 2.5 and Figure 2.6). The components so added are automatically managed. To access this page, follow the Infrastructure -> Components -> Add/Modify menu sequence.

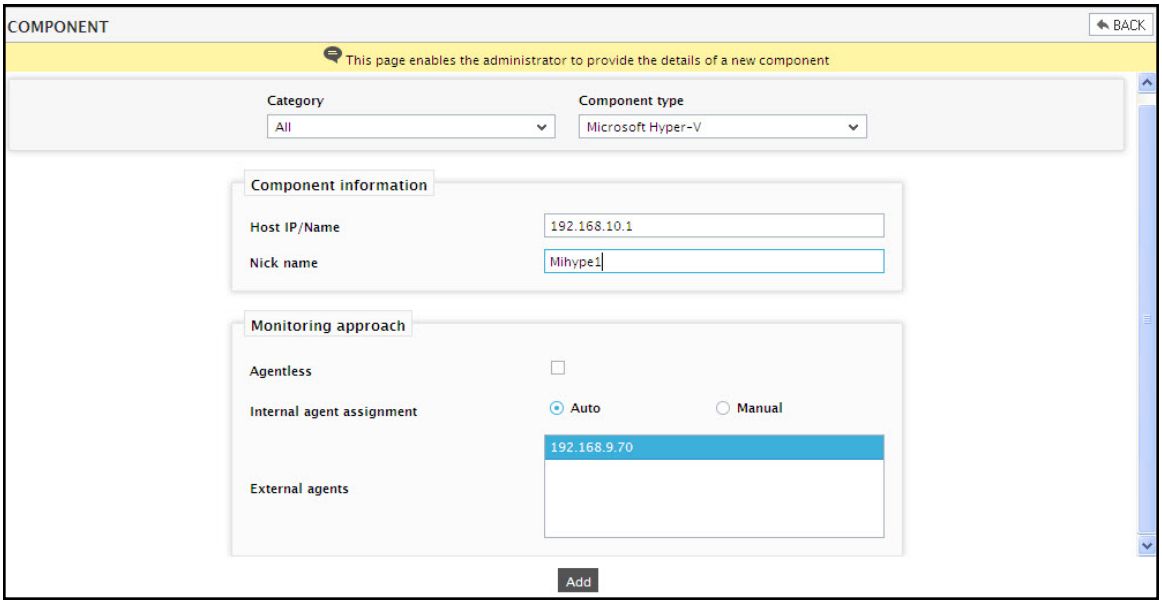


Figure 2.5: Adding the Hyper-V Server

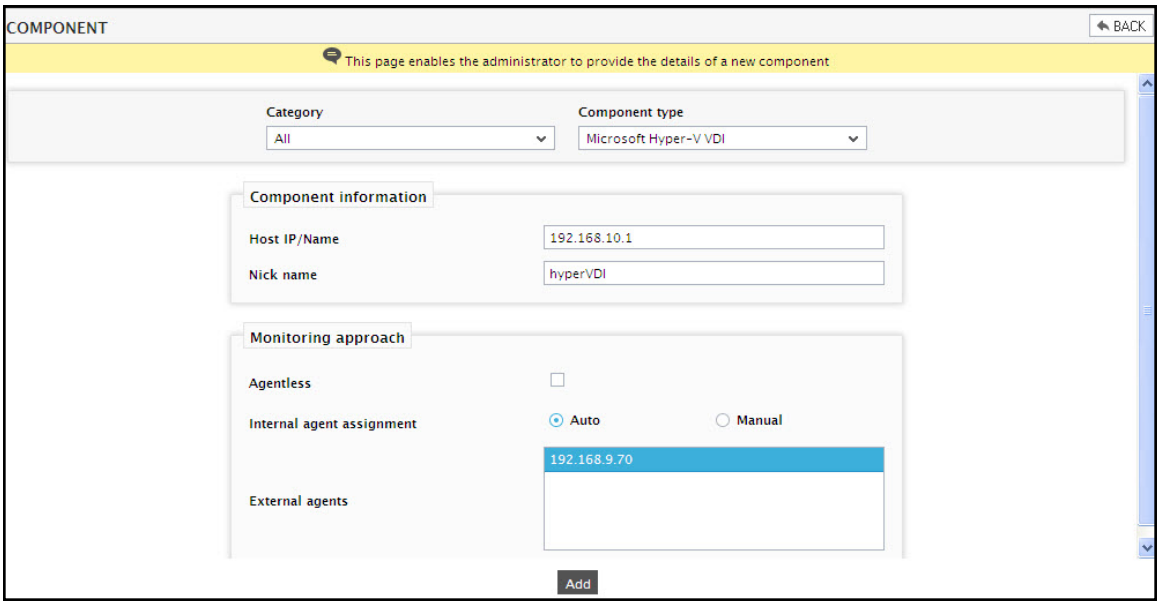


Figure 2.6: Adding the Hyper-V VDI Server

4. When you attempt to sign out, for a Hyper-V server, a list of unconfigured tests appears as in Figure 2.7. For a Hyper-V VDI server, a list of unconfigured tests.

List of unconfigured tests for 'Microsoft Hyper-V'

Performance		Mihype1
Disk Activity - VM	Disk Space - VM	Handles Usage - VM
Hyper-V VM Details	Hyper-V VM Information	Hyper-V VM Snapshots
Memory Usage - VM	System Details - VM	TCP - VM
TCP Traffic - VM	Uptime - VM	Windows Network Traffic - VM
Windows Services - VM		

Figure 2.7: List of tests to be configured for Hyper-V Server

List of unconfigured tests for 'Microsoft Hyper-V VDI'

Performance		hyperVDI
Disk Activity - VM	Disk Space - VM	Handles Usage - VM
Hyper-V Logins	Hyper-V VM Details	Hyper-V VM Information
Hyper-V VM Snapshots	Memory Usage - VM	Personal vDisk - VM
System Details - VM	TCP - VM	TCP Traffic - VM
Uptime - VM	VDI Applications	Virtual Desktop Client's Network Connection
Windows Network Traffic - VM	Windows Services - VM	

Figure 2.8: List of tests to be configured for Hyper-V VDI Server

5. Click on the Hyper-V VM Information test to configure it.

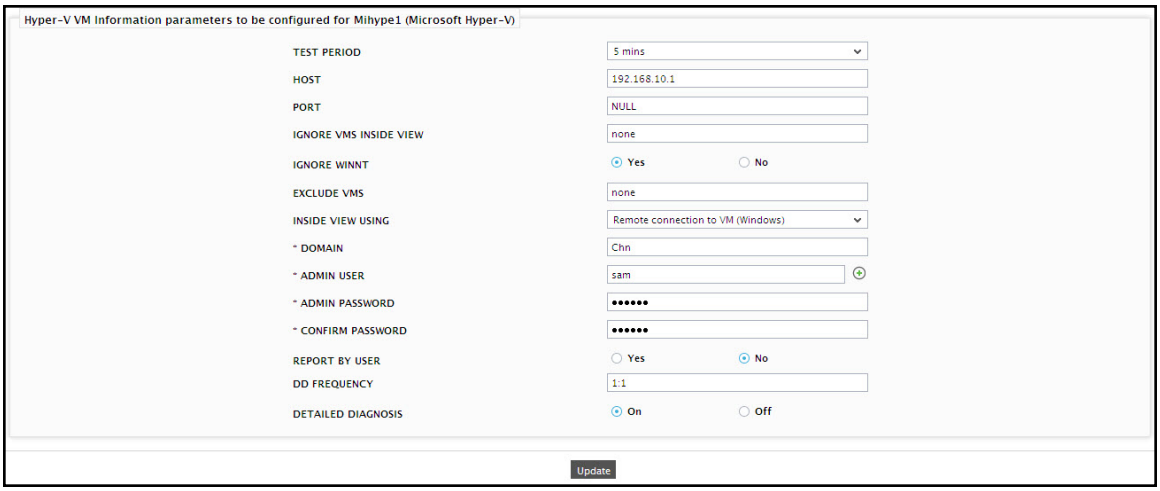


Figure 2.9: Configuring the Hyper-V VM Information test

6. To know how to configure the test, refer to the topic on **Hyper-V VM Information** test.
7. After configuring the test, click on the **Update** button in Figure 2.9.
8. Next, signout of the eG administrative interface.
9. Finally, login to the eG monitoring console to view the current status of the Hyper-V / Hyper-V VDI server. You can zoom into the managed Hyper-V / Hyper-V VDI server and view its layer model, tests, and measurements in the console. For more details on the layer model, refer to the topics on Hyper-V and Hyper-V VDI monitoring models.

The Hyper-V Monitoring Model

eG Enterprise prescribes a specialized *Hyper-V* model (see Figure 3.1) for monitoring Microsoft Hyper-V servers with VMs that host server applications.

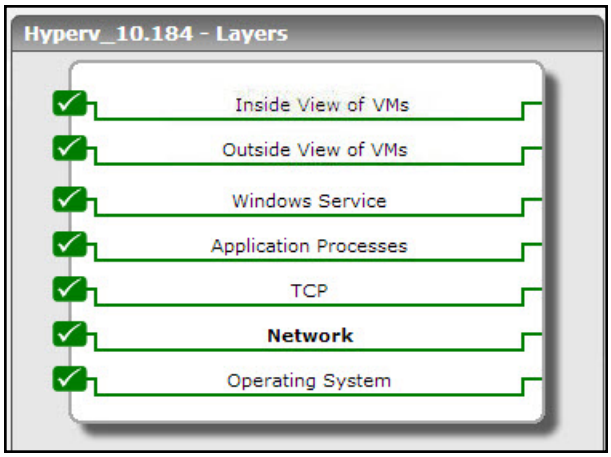


Figure 3.1: The layer model of the Hyper-V server

Each layer of Figure 3.1 execute tests that report on key performance parameters pertaining to the *Hyper-V* server. Using these metrics, administrators can find quick and accurate answers for the following questions:

- Are adequate memory and disk resources available on the Hyper-V host operating system?
- Which logical processor is being used excessively? Who is making more use of the CPU resources – the VMs or the hypervisor?
- How many virtual processors does the hypervisor support? How many of these are available to the root partition? Are all processors used optimally, or have abnormal usage trends been detected with any processor? Which one is it? What is eroding the CPU resources - the guest code or hypervisor code?
- How many memory pages have been deposited with the root partition?
- Do too many TLB flushes occur on the root partition?
- Are too many TLB large pages been used by the root partition?
- How frequently has the root partition attempted to access a page that is not in the CPU TLB?
- How busy is the root partition? Are hypercalls issued by or instructions completed on the root partition very frequently?
- Is the hypervisor managing memory resources efficiently?
- Is the VMBus able to process interrupts smoothly?
- Which is the busiest network adapter/switch/switch port on the Hyper-V server, in terms of amount of network traffic handled?
- Are all critical Hyper-V processes/services available?

- How are the VMs using each virtual processor assigned them? Is any VM over-utilizing the virtual processors?
- Is any VM currently powered off?
- How many VMs are registered with the Hyper-V servers?
- How much physical memory and disk resources have been allocated to every VM? Which VM has been allocated the maximum memory, CPU, and disk resources?
- Are the network adapters supported by the VM healthy or is too much data being lost on the network adapters?
- Which is the busiest VM on the Hyper-V server, in terms of hypercalls and instructions issued/completed?
- Were any VMs migrated to/from the server? If so, which ones are they, and why were they migrated?
- Are any VMs being deleted on the server?
- Are all VMs currently running, or has any VM been paused for a short while?
- Is any VM inaccessible to users?
- How are the VMs using the allocated resources?
- Is any VM currently experiencing a resource crunch? Are any resource-intensive applications/processes executing on that VM?

The layers depicted by Figure 3.1 and the tests mapped to each are discussed elaborately in the sections to come.

3.1 The Operating System Layer

The tests pertaining to the **Operating System** layer (see 3.1) report on the physical resource usage by the Hyper-V host – i.e., the root partition in particular. The physical disk drive that is experiencing excessive activity, the disk drive that is low on space, logical/virtual processors that are over-utilized can be identified accurately using the tests mapped to this layer.

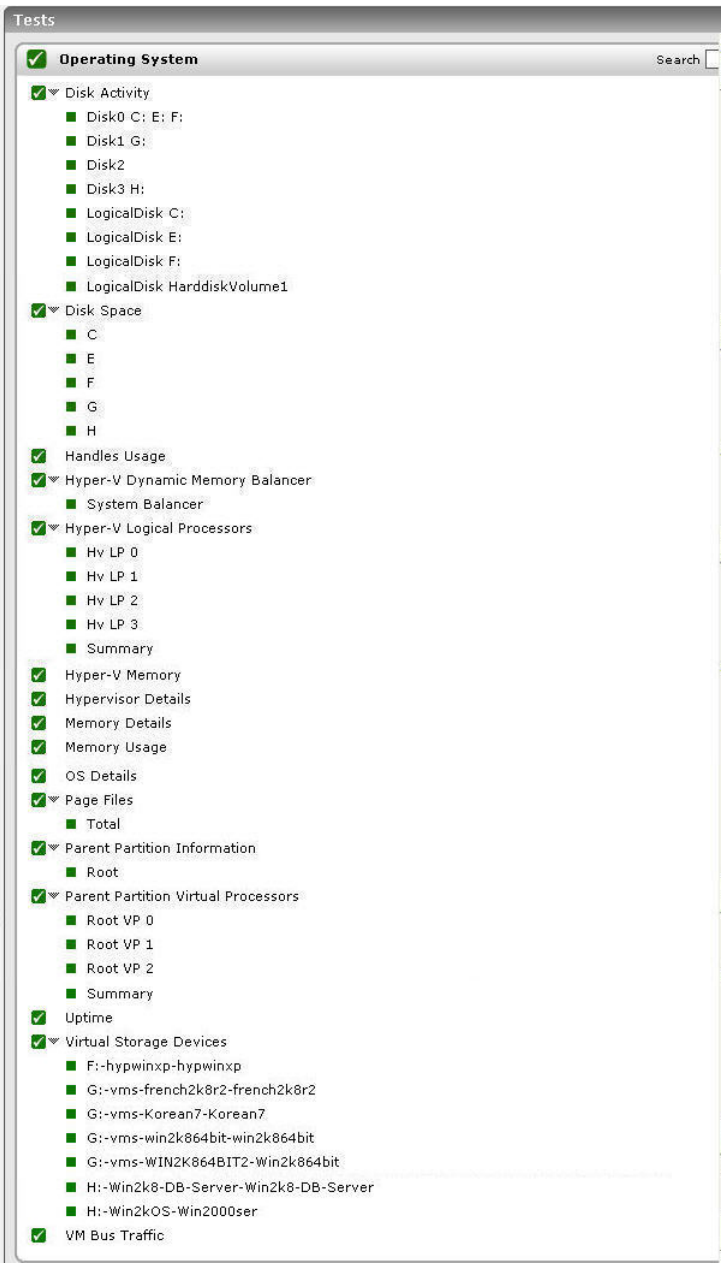


Figure 3.2: The tests mapped to the Operating System layer

We will be discussing the Hyper-V-specific tests only.

3.1.1 Hyper-V Memory Test

This test reports how the Hyper-V host uses the physical memory resources available to it, and reveals whether adequate free memory is available on the host or not.

Target of the test:

A Hyper-V server

Agent executing the test:

An internal agent

Output of the test:

One set of results for the Hyper-V host monitored

Parameters of the test:

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured.
3. **ENABLE MEMORY DIAGNOSIS** - By default, this flag is set to **No**, indicating that detailed diagnosis will not be available for the Free memory measure reported by this test by default. If you want to view the detailed diagnosis of the Free memory measure - i.e., to view the top 10 processes on the Hyper-V host that are utilizing memory excessively - you can change this flag to **Yes**.
4. **DETAILED DIAGNOSIS** - To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option.

   The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:

   - The eG manager license should allow the detailed diagnosis capability
   - Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

   The ability of this test to provide detailed diagnostic measures is governed by the enable memory diagnosis parameter only. This test therefore, disregards the status of the detailed diagnosis flag when determining whether/not to collect detailed metrics.

Measures reported by the test:

Measurement Name	Description	Measurement Unit	Interpretation
Run queue length	Indicates the instantaneous	Number	A value consistently greater than 2 indicates that many processes

Measurement Name	Description	Measurement Unit	Interpretation
	length of the queue in which threads are waiting for the processor cycle. This length does not include the threads that are currently being executed.		could be simultaneously contending for the processor.
Number of blocked processes	Indicates the number of processes currently blocked for I/O, paging, etc.	Number	A high value could indicate an I/O problem on the host (e.g., a slow disk).
Swap memory	This measurement denotes the committed amount of virtual memory. This corresponds to the space reserved for virtual memory on disk paging file(s).	MB	An unusually high value for the swap usage can indicate a memory bottleneck. Check the memory utilization of individual processes to figure out the process(es) that has (have) maximum memory consumption and look to tune their memory usages and allocations accordingly.
Free memory	Indicates the free memory available currently.	MB	A very low value of free memory is also an indication of high memory utilization on a host. The detailed diagnosis of this measure lists the top 10 processes responsible for maximum memory consumption on the host.
Scan rate	Indicates the memory scan rate.	Pages/Sec	A high value is indicative of memory thrashing. Excessive thrashing can be detrimental to application performance.

The detailed diagnosis of the *Free memory* measure lists the top-10 memory-consuming processes on the Hyper-V host. Using this information, you can accurately identify the process that is causing the memory drain on the host.

Lists the top 10 memory processes

Time	PID	%MEM	ARGS
Mar 13, 2009 16:00:27			
	1616	13.68	js
	384	1.8	svchost
	2348	1.78	vmms
	2992	0.89	vmwp
	3012	0.85	vmwp
	3004	0.76	vmwp
	1836	0.75	svchost
	1020	0.71	svchost
	3320	0.65	explorer
	3916	0.65	mmc

Figure 3.3: The top 10 memory consumers on the Hyper-V host

3.1.2 Hyper-V Memory Usage Test

If one/more VMs on a Hyper-V host are over-sized with physical memory resources, it can result in a serious memory contention that may not only affect the host, but also other VMs on the host. By tracking the physical memory allocation to the VMs on a Hyper-V host, administrators can proactively detect over-allocations and can initiate remedial actions before the problem impacts performance. For this, administrators can use the **Hyper-V Memory Usage** test. This test monitors the physical memory allocated to the VMs, and points to those VMs that are allocated more resources than required.

Target of the test:

A Hyper-V / Hyper-V VDI server

Agent executing the test:

An internal agent

Output of the test:

One set of results for the Hyper-V host monitored

Parameters of the test:

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured.
3. **DETAILED DIAGNOSIS** - To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the

detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option. The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:

- The eG manager license should allow the detailed diagnosis capability
- Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
Total memory available	Indicates the total physical memory capacity of the host.	MB	
Memory allocated to VMs	Indicates amount of physical memory allocated to VMs.	MB	
Memory used by VMs	Indicates the percentage of physical memory that is allocated to the VMs on the host.	Percent	Ideally, this value should be low. A value close to 100% is indicative of over-allocation of physical memory to the VMs. You can use the detailed diagnosis of this measure to identify which VM has been allocated maximum resources.
Available memory for VMs	Indicates the amount of physical memory unused on the host.	MB	Ideally, the value of this measure should be high. A low value or a consistent decrease in this value is a cause for concern, as it indicates excessive memory usage by the VMs on the host.

3.1.3 Hyper-V Logical Processors Test

A logical processor is a hardware entity, either a processor core or a hyperthread, on which the Hyper-V operating system can schedule a software thread for execution.

The metrics reported by this test enables administrators to figure out the following:

- Which logical processor is being used excessively? Who is making more use of the CPU resources – the VMs or the hypervisor?
- Are the logical processors able to process interrupts well? Is any logical processor experiencing a bottleneck during interrupt processing?
- Has any processor been idle for too long a time? Does that processor receive scheduler interrupts frequently?

Target of the test:

A Hyper-V / Hyper-V VDI server

Agent deploying the test:

An internal agent

Output of the test:

One set of results for every logical processor on the Hyper-V host monitored

Parameters of the test:

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
Guest runtime	Indicates the percentage of time guest code is running on this logical processor (LP). For the Summary descriptor, it indicates the average percentage across all logical processors.	Percent	For example, if you have 2 logical processors and one VM running CPU tests you might see the value be 95% for LP (0), 0% for LP (1) and 47.5% for the Summary. From this, you can conclude which processor is being heavily used by the VMs.

Measurement	Description	Measurement Unit	Interpretation
Hypervisor runtime	Indicates percentage of time the Hypervisor is running on an LP. For the Summary descriptor, this measure indicates the average percentage across all LPs.	Percent	Ideally, this value should be low. Comparing the value of this measure across LPs will enable you to accurately identify the LP that is being excessively used by the hypervisor.
Idle time	Indicates the percentage of time the LP is waiting for work. For the Summary descriptor, this measure indicates the average percentage across all LPs.	Percent	Ideally, this value should be low. Comparing the value of this measure across LPs will enable you to accurately identify the LP that is the most idle.
CPU utilization	Indicates the percentage of time this LP was in use.	Percent	This is typically the sum of the Guest runtime and Hypervisor runtime measures. Comparing the value of this measure across LPs will reveal the LP that is being utilized excessively. If the value of this measure is less than 60% consumed, then the LP usage is considered Healthy. A usage level between 60% and 89% consumed, can be considered as a warning. A value between 90% and 100% is indicative of a serious resource contention.
Context switches	Indicates the number of times per second a new Virtual Processor (VP) had been scheduled to a particular Logical Processor (LP). For the **Summary** descriptor, the value of this measure indicates the total number of VP to LP switches per second.	Switches/Sec	Ideal time context switches of around 1000 for a single guest running are not uncommon. This is due to the fact the VP will "Halt" and allow something else to run if it has no work to do.

Measurement	Description	Measurement Unit	Interpretation
Hardware interrupts	Indicates the number of hardware interrupts this LP is processing per second. For the Summary descriptor, the value of this measure is the total number of hardware interrupts per second across all LPs.	Interrupts/Sec	Hardware interrupts are delivered to the root VP's corresponding the LP on which it was received. For example a network card will create and interrupt when a packet is received.
Inter processor interrupts received	Indicates the total number of Inter-processor interrupts (IPI) received per second of a given LP. For the **Summary** descriptor, this is the total number of IPIs received by all LPs.	Interrupts/Sec	IPIs are sent from one processor to another to get the processor to do memory coherency (like TLB, cache, etc.).
Inter processor interrupts sent	Indicates the number of Ips sent per second of a given LP. For the Summary descriptor, this is the total number of IPIs sent by all LPs.	Interrupts/Sec	
Monitor transition cost	This is a current measure of the cost to enter the Hypervisor via an Intercept on a Logical Processor (LP). For the **Summary** descriptor, it is the total cost across all processors.	Number	Intercepts are like User mode to Kernel Mode context switches except that here it is from the User/Kernel Mode to the Virtual Machine Monitor (VMM) a.k.a the Hypervisor mode. The smaller this value the better. The only real use it has is to figure out the relative performance of processors.
Scheduler interrupts	Indicates the number of scheduler interrupts that occurred on this LP per second. For the **Summary** descriptor, the number of scheduler interrupts that occurred across all LPs will be reported as the	Interrupts/Sec	Scheduler interrupts are sent by the Hypervisor scheduler from one Logical Processor (LP) to another to re-evaluate their runlist. The runlist is the list of Virtual Processors (VP) waiting

Measurement	Description	Measurement Unit	Interpretation
	value of this measure.		to run on a given LP. This is also a “wake-up” mechanism for an LP that might be sitting idle in a lower power state.

3.1.4 Parent Partition Information Test

The hypervisor creates partitions that are used to isolate guests and host operating systems. A partition is comprised of a physical address space and one or more virtual processors. A parent partition creates and manages child partitions. It contains a virtualization stack, which controls these child partitions. The parent partition is in most occasions also the root partition. It is the first partition that is created and owns all resources not owned by the hypervisor. As the root partition it will handle the loading of and the booting of the hypervisor. It is also required to deal with power management, plug and play and hardware failure events.

The Parent Partition Information test monitors the root partition and reports how well the root manages the physical memory resources.

Target of the test:

A Hyper-V / Hyper - V VDI server

Agent executing the test:

An internal agent

Output of the test:

One set of results for the root partition on the Hyper-V host monitored

Parameters of the test:

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
Address space in the virtual TLB	Indicates the current number of address spaces in the virtual TLB of the partition.	Number	The root partition hosts a data structure called a Translation LookAside Buffer (TLB), which is used to perform Virtual to Physical Address Translation. A TLB has a fixed number of slots containing page table entries, which map virtual addresses onto physical addresses. It is typically a content-addressable memory (CAM), in which the search key is the virtual address and the search result is a physical address. This measure is a good indicator of the size of the TLB.
Virtual processors	Indicates the number of virtual processors present in the root partition currently.	Number	All execution in the root and child partitions happens on Virtual Processors (VPs). At a minimum you will see one VP for each Logical Processor (LP). These account for the root VPs.
Deposited pages	Indicates the number of pages currently deposited into this partition.	Number	For each partition, the hypervisor maintains a memory pool of RAM SPA pages. This pool acts just like a checking account. The amount of pages in the pool is called the *balance*. Pages are *deposited* or *withdrawn* from the pool. When a hypercall that requires memory is made by a partition, the hypervisor withdraws the required memory from the total pool balance of that partition.
Pages used by the virtual TLB	Indicates the number of page tables that are currently	Number	

Measurement	Description	Measurement Unit	Interpretation
	present in the virtual TLB of the partition.		
Pages present in the GPA space	Indicates the number of pages currently present in the GPA space of the root partition.	Number	The physical memory that is seen by the hypervisor is called System Physical Address (SPA) space. The pages allocated for the operating system in a child partition are not necessarily contiguous so a remapping takes place to allow the guest to see a contiguous Guest Physical Address (GPA) space. System Physical Address space refers to the physical memory's physical addresses. Guest Physical Address space is the set of pages that are accessed when a guest references a physical address (i.e. when the CR3 register is loaded with the physical address of the page directory). There is one SPA space per machine and one GPA space per child partition. When the operating system is running within a child partition using Hyper-V, the guest page tables reference GPA, although, as far as the child partition operating system knows, this is physical memory. Even though the guest references a GPA page when referring to physical memory, these references have to be converted so the actual memory access is performed on an SPA page.
GPA space modification	Indicates the rate of modifications to the GPA space.	Modifications/Sec	

Measurement	Description	Measurement Unit	Interpretation
Virtual TLB flush entries	Indicates the rate of flushes of the entire virtual TLB.	Entries/Sec	When the memory map is changed the entries in the TLB many need to be removed (flushed). TLB flushes can be expensive operations because it may trigger interprocessor interrupts to clear out similar entries on other processors and additional accesses to memory to recompute mappings that were previously in the TLB. Therefore, the value of this measure, should ideally be low.

3.1.5 Parent Partition Virtual Processors Test

A virtual processor is a single logical processor that is exposed to a partition by the hypervisor. Virtual processors can be mapped to any of the available logical processors in the physical computer and are scheduled by the hypervisor to allow you to have more virtual processors than you have logical processors.

This test monitors how well the parent partition uses the virtual processors assigned to it.

Target of the test:

A Hyper-V / Hyper-V VDI server

Agent executing the test:

An internal agent

Output of the test:

One set of results for the every virtual processor assigned to the root partition of the Hyper-V host monitored

Parameters of the test:

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
Parent partition run time	Indicates the percentage of time spent by this virtual processor (VP) in guest code. For the Summary descriptor, the value of this measure is the total percentage across all VPs.	Percent	Comparing the value of this measure across VPs will accurately indicate which VP is being actively used by the guests.
Hypervisor runtime	Indicates the percentage of time spent by the virtual processor in hypervisor code. For the Summary descriptor, the value of this measure is the total percentage across all VPs.	Percent	Comparing the value of this measure across VPs will accurately indicate which VP is being actively used by the hypervisor.
Parent partition CPU utilization	Indicates the total percentage of time this VP was in use. For the Summary descriptor, this is the average percentage of time for which all VPs were in use.	Percent	This is typically the sum of the *Parent partition runtime* and *Hypervisor runtime* measures. Comparing the value of this measure across VPs will reveal the VP that is being utilized excessively.
Control register accesses	Indicates the number of CPU Control Register accesses per second. For the Summary descriptor, this is rate of CPU control register accesses across all VPs.	Accesses/Sec	Control registers are used to set up address mapping, privilege mode, etc.
CPUID instructions	Indicates the number of CPUID instructions calls per second. For the Summary descriptor, this is rate of CPUID instructions across all VPs.	Instructions/Sec	The CPUID instruction is used to retrieve information on the local CPU's capabilities. Typically, CPUID is only called when the OS /

Measurement	Description	Measurement Unit	Interpretation
			Application first start. Therefore, this value is likely to be 0 most of the time.
Emulated instructions	Indicates the number of emulated instructions completed per second. For the Summary descriptor, this is rate of emulated instructions completed across all VPs.	Instructions/Sec	Some instructions require emulation to complete in the Hypervisor. One such example is APIC access.
HLT instructions	Indicates the number of CPU halts per second on the VP.For the Summary descriptor, this is the total number of CPU halts (per second) across all VPs.	Instructions/Sec	A HLT will cause the hypervisor scheduler to de-schedule the current VP and move to the next VP in the runlist.
Hypercalls	Indicates the number of hypercalls made by guest code on the VP per second. For the Summary descriptor, this is the total number of hypercalls made on all VPs per second.	Hypercalls/Sec	Hypercalls are one form of enlightenment. Guest OS's use the enlightenments to more efficiently use the system via the hypervisor. TLB flush is an example hypercall. If this value is zero, it is an indication that Integration Components are not installed. New OS's like WS08 can use hypercalls without enlightened drivers. So, hypercalls are only a prerequisite and not a guarantee for not having Integration Components installed.
IO instructions	Indicates the number of CPU in / out instructionsexecuted per second. For the Summary descriptor, this is total number of IO instructions executed on all VPs per second.	Instructions/Sec	Many older or low bandwidth devices use "programmed I/O" via in / out instructions.

Measurement	Description	Measurement Unit	Interpretation
Large page TLB fills	Indicates the number of Large Page TLB fills / second.For the Summary descriptor, this is rate of large page TLB fills across all VPs per second.	Fills/Sec	There are two types of TLB entries (and some three). Small TLB which generally means a 4K page and large Page which generally means 2MB. There are fewer Large TLB entries on the order of 8 – 32. A non- zero value for this measures indicates that the root partition is using large pages.
MSR accesses	Indicates the number of Machine Specific Register (MSR) instruction calls per second.For the Summary descriptor, this is total number of MSR instruction calls made on all VPs per second.	Accesses/Sec	There are many types of MSRs such as C-state config, Synthetic Interrupt (Synic) Timers, and control functions such as shutdown.
MWAIT instructions	Indicates the number of MWAIT instructions per second. For the Summary descriptor, this is the total number of MWAIT instructions executed on all VPs per second.	Instructions/Sec	The mwait (monitored wait) instruction instructs the processor to enter a wait state in which the processor is instructed to monitor the address range between a and b and wait for an event or a store to that address range.
Page fault intercepts	Indicates the number of page faults per second. For the Summary descriptor, this is the total number of page faults on all VPs per second.	Intercepts/Sec	Whenever guest code accesses a page not in the CPU TLB a page fault will occur. This counter is closely correlated with the Large Page TLB Fills measure.
Small page TLB fills	Indicates the number of Small Page TLB fills / second. For the Summary descriptor, this is rate of small page	Fills/Sec	There are two types of TLB entries (and some three). Small TLB which generally

Measurement	Description	Measurement Unit	Interpretation
	TLB fills across all VPs.		means a 4K page and large Page which generally means 2MB. There are fewer Large TLB entries on the order of 64 – 1024+.
Total intercepts	Indicates the rate of hypervisor intercept messages. For the **Summary** descriptor, this is rate at which intercepts occurred across all VPs per second.	Intercepts/Sec	Whenever a guest VP needs to exit its current mode of running for servicing in the hypervisor, this is called an intercept. Some common causes of intercepts are resolving Guest Physical Address (GPA) to Server Physical Address (SPA) translations, privileged instructions like hlt / cupid / in / out, and the end of the VP's scheduled time slice.

3.1.6 VM Bus Traffic Test

Child partitions do not have direct access to hardware resources, but instead have a virtual view of the resources, in terms of virtual devices. Any request to the virtual devices is redirected via the VMBus to the devices in the parent partition, which will manage the requests. The VMBus is a logical channel which enables inter-partition communication. The response is also redirected via the VMBus.

Using this test, you can measure the level of activity on the VMBus.

Target of the test:

A Hyper-V / Hyper-V VDI server

Agent executing the test:

An internal agent

Output of the test:

One set of results for the Hyper-V host monitored

Parameters of the test:

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
Interrupts received	Indicates the number of interrupts received since the last measurement period.	Number	
Interrupts sent	Indicates the number of interrupts sent since the last measurement period.	Number	
Throttle events	Indicates the total number of times since the last measurement period that any partition has been throttled, which is to say that its interrupts were disabled.	Number	

3.1.7 Hypervisor Status Test

A hypervisor, also called virtual machine monitor (VMM), is a computer hardware platform virtualization software that allows multiple operating systems to run on a host computer concurrently. It runs directly on the host's hardware as a hardware control and guest operating system monitor. A guest operating system thus runs on another level above the hypervisor.

The Hypervisor Status test reports useful statistics revealing the health of the Hyper-V hypervisor.

Target of the test : A Hyper-V / Hyper-V VDI server

Agent executing the test : An internal agent

Output of the test : One set of results for the Hyper-V host monitored

Configurable parameters for the test

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Logical processors	Indicates the number of cores / HT that the hypervisor is managing currently.	Number	If you have a dual proc quad core without HT you will see this number set to 8. If you also had HT it would be set to 16.
Virtual machines running	Indicates the number of partitions managed by the hypervisor currently.	Number	Each virtual machine on the system is run in a container called a partition. If you have no VMs running this value will be set to 1, because the "host OS" called the "root" in Hyper-V is also running in a partition. So, if you have 2 guest VMs running, this value will be 3 - 2 for each guest VM and 1 for the root.
Virtual processors	Indicates the number of virtual processors on the system currently.	Number	All execution in the root and child partitions (where guest VMs run) happens on Virtual Processors (VPs). At a minimum, you will see one VP for each Logical Processor (LP). These account for the root VPs. You will then see one for each VP you have configured to a guest. Therefore, if you have an 8LP system with 1 guest running with 2 VPs, the count here will be 10.
Monitored notification	Indicates the number of monitored notifications currently registered with the hypervisor.	Number	Monitored notifications are part of an interrupt coalescing technique Hyper-V uses to reduce virtualization overhead. For example, when a guest has data to transmit over the network it could send an interrupt for each packet to the root VP that will actually do the I/O, or it can

Measurement	Description	Measurement Unit	Interpretation
			send one interrupt to let the root know data is starting to flow. This counter is an indication of the number of "flows" of interrupts being set to the root and guests.
Total pages	Indicates the current number of bootstrap and deposited pages in the hypervisor.	Number	The Hypervisor needs memory in order to keep track of Virtual Processors, Guest Virtual address to System Physical Address translation entries in the virtual TLB, etc. Therefore, the total pages keep track of the total amount of memory the Hypervisor is using for management or partitions. A page is 4KBytes. This is not the total amount used to support a guest. You would also need to get this by looking at the size of the worker process (vmwp.exe) and account for memory in vid. Total Pages can change based on what guests VMs are running. Here is an example of how the Hypervisor gets memory - A user want to start a VM. To achieve this, the vid makes a hypercall to the Hypervisor via winhv.sys to create a partition. In order to create VPs, vTLBs, etc., the Hypervisor needs memory. Hence, it makes a call to the root via winhv.sys. Winhv.sys then allocates memory from the root and makes a hypercall to deposit memory and then the whole process unwinds and the partition create completes.

3.1.8 Hyper-V Dynamic Memory Balancer Test

Dynamic Memory is a new feature of Hyper-V™ that enables Hyper-V hosts to dynamically adjust the amount of memory available to virtual machines in response to changing workloads. Instead of assigning a specific

amount of memory to a virtual machine, the administrator instead configures a range of memory, memory priority and other settings that Hyper-V then uses to determine how much memory to allocate to the virtual machine in real time. The benefits of Dynamic Memory include higher virtual machine consolidation ratios and increased flexibility for managing virtualized workloads.

By closely monitoring the amount of memory the Hyper-V host dynamically allocates and releases from VMs, you can understand the memory needs of virtual machines and the memory pressure on the host. The Hyper-V Dynamic Memory Balancer test enables this monitoring and the consequent analysis.

Target of the test:

A Hyper-V / Hyper-V VDI server

Agent executing the test:

An internal agent

Output of the test:

One set of results for the Hyper-V host monitored

Parameters of the test:

1. Test period - How often should the test be executed
2. **HOST** - The host for which the test is to be configured.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
Added memory	Indicates the amount of physical memory added to the VMs during the last measurement period.	MB	Hyper- V host and the enlightened VM communicate through the VMBus (the server use Virtual Service Provider and the client use Virtual Service Consumer) to determine the current memory needs of the VM. If the workload of the VM increases and need more

Measurement	Description	Measurement Unit	Interpretation
			memory – then memory is dynamically added to the VM. If the workload decreases (or other VMs have higher memory priority)– the memory is dynamically removed from the VM.
Removed memory	Indicates the amount of physical memory removed from the VMs during the last measurement period.	MB	
Available memory	Indicates the amount of physical memory remaining unused on the host.	MB	A very low value of this measure is a cause of concern. This is because, it indicates that the physical memory of the host has been overcommitted; if too much paging occurs at this juncture, performance will plummet.
Average pressure	Indicates the average memory pressure on the host.	Percent	Dynamic Memory determines the amount of memory needed by a virtual machine by calculating something called memory pressure. To perform this calculation, Hyper-V looks at the total committed memory of the guest operating system running in the virtual machine and then calculates pressure as the ratio of how much memory the virtual machine wants to how much it has. The amount of memory that Hyper- V then assigns to the virtual machine equals total committed memory plus some additional memory to be used as a buffer. However, Dynamic Memory does not guarantee that the total committed memory amount is always assigned to the virtual

Measurement	Description	Measurement Unit	Interpretation
			machine. Neither does Dynamic Memory guarantee that the additional memory amount configured as a buffer value is always assigned to the virtual memory. This is because the actual amount of memory assigned to a virtual machine depends upon the memory pressure being exerted upon the host by the memory needs of other virtual machines running on the host. A very high value of this measure therefore indicates that the VMs are exerting too much memory pressure on the host, probably owing to a severe memory contention on the VMs. As long as this number is under 100, you can conclude that there is enough memory on the Hyper-V host to service your virtual machines. Ideally, this value should be at 80 or lower. The closer this gets to 100, the closer you are to running out of memory. Once this number goes over 100 then you can pretty much guarantee that you have virtual machines that are paging in the guest operating system.
Memory add operations	Indicates the number of memory addition operations performed during the last measurement period.	Number	
Memory remove	Indicates the number of memory	**Number**	

Measurement	Description	Measurement Unit	Interpretation
operations	removal operations performed during the last measurement period.		

3.1.9 Virtual Storage Devices Test

Virtual hard disk (VHD) is a disk image file format for storing the complete contents of a hard drive. It replicates an existing hard drive and includes all data and structural elements.

On the Hyper-V management operating system, virtual hard disks can have a capacity of up to 2040 gigabytes and can be of any of the following types:

- Fixed: A fixed virtual hard disk is a disk that occupies physical disk space on the management operating system equal to the maximum size of the disk, regardless of whether a virtual machine requires the disk space. A fixed virtual hard disk takes longer to create than other types of disks because the allocated size of the .vhd file is determined when it is created. This type of virtual hard disk provides improved performance compared to other types because fixed virtual hard disks are stored in a contiguous block on the management operating system.
- Dynamically expanding: A dynamically expanding virtual hard disk is a disk in which the size of the .vhd file grows as data is written to the disk. This type provides the most efficient use of disk space. You will need to monitor the available disk space to avoid running out of disk space on the management operating system.
- Differencing: A differencing virtual hard disk stores the differences from the virtual hard disk on the management operating system. This allows you to isolate changes to a virtual machine and keep a virtual hard disk in an unchanged state. The differencing disk on the management operating system can be shared with virtual machines and, as a best practice, must remain read-only. If it is not read-only, the virtual machine‘s virtual hard disk will be invalidated.

The performance of a virtual hard disk is judged by the speed with which it processes I/O requests. Slowdowns in I/O processing can cause read-write requests to the virtual hard disks to queue up, thereby increasing the I/O load on the virtual hard disks and degrading the overall performance of the VMs using those virtual hard disks. Moreover, since virtual hard disk files (.vhd files) are typically stored in the physical disks of the Hyper-V host, excessive I/O activity on the virtual hard disk will also impact the performance of the corresponding physical disks.

Using the **Virtual Storage Devices** test, administrators can periodically monitor the I/O activity on each virtual hard disk assigned to every VM on Hyper-V. This way, probable delays in the processing of I/O requests can be proactively detected, and the physical disks and VMs that may be impacted by these latencies can be isolated.

Target of the test : A Hyper-V / Hyper-V VDI server

Agent executing the test : An internal agent

Output of the test : One set of results for each virtual disk hosted by a physical disk and assigned to a VM

Parameters of the test:

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
Read throughput	Indicates the rate at which data is read from this virtual hard disk that is stored by this physical disk and used by this VM.	Bytes/Sec	While an abnormal increase in the value of these measures could indicate a high level of read-write activity on a virtual hard disk, a consistent decrease in the value of these measures could be indicative of a processing bottleneck probably caused by slowdowns in a virtual hard disk. When a physical disk experiences slowdowns, you can compare the value of these measures across all virtual hard disks that are located on that physical disk, so that you can instantly identify which virtual hard disk is contributing to the processing delay.
Write throughput	Indicates the rate at which data is written to this virtual hard disk that is stored by this physical disk and used by this VM.	Bytes/Sec	
Total throughput	Indicates the rate at which data is read from/written to this virtual hard disk that is stored by this physical disk and used by this VM.	KBps	
Read IOPS	Indicates the rate at which data reads are performed on this virtual hard disk that is stored by this physical disk and used by this VM.	Reads/sec	
Write IOPS	Indicates the rate at which data writes are performed on this virtual hard disk that is stored by this physical disk and used by this VM.	Writes/sec	

Measurement	Description	Measurement Unit	Interpretation
Total IOPS	Indicates the total number of data reads and data writes performed per second on this virtual hard disk that is stored by this physical disk and used by this VM.	Operations/sec	
Errors count	Indicates the number of error events triggered on this virtual hard disk that is stored by this physical disk and used by this VM.	Number	
Flush count		Number	
Read count	Indicates the number of read operations performed on this virtual hard disk that is stored by this physical disk and used by this VM.	Number	
Write count	Indicates the number of write operations performed on this virtual hard disk that is stored by this physical disk and used by this VM.	Number	

3.1.10 Hyper-V Memory Reserve Test

As Hyper-V can dynamically allocate memory to virtual machines on demand, the host needs to ensure that some memory is kept for itself. Without which, Hyper-V may allocate a lot of memory to the VMs, starving the host of adequate memory resources. As a result, the host may start performing poorly. To avoid this, administrators need to time and again check the amount of memory that the host has set aside for its use. This check can be easily performed using the **Hyper-V Memory Reserve** test. This test periodically reports the amount of memory that the host has reserved for itself, thus enabling administrators to periodically check whether the host has sufficient memory for its own operations.

Target of the test:

A Hyper-V / Hyper-V VDI server

Agent executing the test:

An internal agent

Output of the test:

One set of results for the Hyper-V host being monitored

Parameters of the test:

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
Host memory reserve	Indicates the amount of memory reserved for the host.	MB	A high value is desired for this measure. If not, the host's performance may suffer, which in time, will affect the VMs' performance as well.

3.2 The Network Layer

The tests mapped to this layer indicate whether the Hyper-V server is available over the network or not, and if so, how quickly it responds to requests. In addition, the tests measure the level of network traffic handled by the adapters, switches, and switch ports supported by the Hyper-V server.

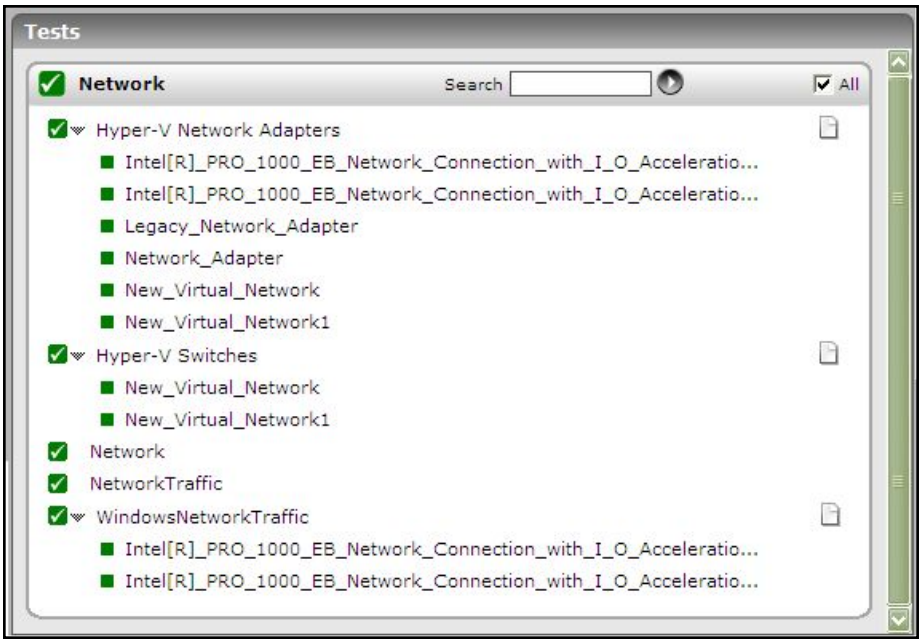


Figure 3.4: The tests mapped to the Network layer

3.2.1 Hyper-V Network Adapters Test

There are two types of network adapters available for Hyper-V: a network adapter and a legacy network adapter. For the network adapter to work, integration services must be installed, which is part of the Hyper-V installation. If integration services cannot be installed because of the version of the operating system, the network adapter cannot be used. Instead, you need to add a legacy network adapter that emulates an Intel 21140-based PCI Fast Ethernet Adapter and works without installing a virtual machine driver. A legacy network adapter also supports network-based installations because it includes the ability to boot to the Pre-Execution Environment (PXE boot). The legacy network adapter is also required if a virtual machine needs to boot from a network. You will need to disable the network adapter after the PXE boot.

Hyper-V allows guest computers to share the same physical network adapter. It is therefore necessary for administrators to monitor how each of the physical network adapters are used by both the guest VMs and the host operating system, so that they can accurately determine which adapter is experiencing high usage levels, and also figure out where the network resources are spent more – at the VM-level or at the host operating system-level?

To determine how the Hyper-V operating system utilizes these network adapters, periodically execute the Hyper-V Network Adapters test.

Target of the test : A Hyper-V / Hyper-V VDI server

Agent executing the test : An internal agent

Output of the test : One set of results for each network adapter available to the Hyper-V host monitored

Configurable parameters for the test

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Broadcast packets received	Indicates the total number of packets received per second by this adapter.	Packets/Sec	
Broadcast packets sent	Indicates the total number of packets sent per second by this network adapter.	Packets/Sec	
Data received	Indicates the rate at which bytes were received by this network adapter.	Mbps	
Data transmitted	Indicates the rate at which bytes of data were sent by this network adapter.	Mbps	
Packets received	Indicates the total number of packets received by this network adapter per second.	Packets/Sec	
Packets sent	Indicates the total number of packets sent by this network adapter per second.	Packets/Sec	
Is enabled?	Indicates whether/not this network adapter is enabled.		The values that this measure reports and the numeric values that correspond to them have been discussed in the table below:

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><th>State</th><th>Numeric Value</th></tr>
<tr><td>No</td><td>0</td></tr>
<tr><td>Yes</td><td>1</td></tr>
</table>
Note:
By default, this measure reports the Measure Values listed in the table above to indicate whether/not this network adapter is enabled. The graph of this measure however, represents the same using the numeric equivalents only.
</td></tr>
<tr><td>Uplink status:</td><td>Indicates the current uplink status of this network adapter.</td><td></td><td>The values that this measure reports and the numeric values that correspond to them have been discussed in the table below:
<table>
<tr><th>State</th><th>Numeric Value</th></tr>
<tr><td>Down</td><td>0</td></tr>
<tr><td>Degraded</td><td>1</td></tr>
<tr><td>Up</td><td>2</td></tr>
</table>
Note:
By default, this measure reports the Measure Values listed in the table above to indicate the current uplink status of this network adapter. The graph of this measure however, represents the same using the numeric equivalents only.
</td></tr>
</table>

3.2.2 Hyper-V Switches Test

A virtual switch can be attached to one and only one physical NIC. Each Virtual / Legacy NIC plugs into a virtual switch. This test gives details on what the switch is doing and the flows of sends / receives it handles.

Target of the test:

A Hyper-V / Hyper-V VDI server

Agent executing the test:

An internal agent

Output of the test:

One set of results for each virtual switch available to the Hyper-V host monitored

Parameters of the test:

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
Broadcast packets received	Indicates the total number of packets received per second by this switch.	Packets/Sec	
Broadcast packets sent	Indicates the total number of packets sent per second by the switch.	Packets/Sec	
Data received	Indicates the rate at which bytes were received by the switch.	Mbps	
Data transmitted	Indicates the rate at which bytes of data were sent by the switch.	Mbps	
Packets received	Indicates the total number of packets received by the switch	Packets/Sec	

Measurement	Description	Measurement Unit	Interpretation
	per second.		
Packets sent	Indicates the total number of packets sent by the switch per second.	Packets/Sec	

3.2.3 Hyper-V Switch Ports Test

This test reports the network traffic flowing into and out of every virtual switch port (i.e., virtual NIC) on the Hyper-V host. This test is disabled by default. To enable the test, go to the **ENABLE / DISABLE TESTS** page using the menu sequence : Agents -> Tests -> Enable/Disable, pick Hyper-V as the **Component type**, Performance as the **Test type**, choose the test from the **DISABLED TESTS** list, and click on the **>>** button to move the test to the **ENABLED TESTS** list. Finally, click the **Update** button.

Target of the test:

A Hyper-V / Hyper-V VDI server

Agent executing the test:

An internal agent

Output of the test:

One set of results for each virtual switch port available to the Hyper-V host monitored

Parameters of the test:

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
Broadcast packets received	Indicates the total number of packets received per second by this port.	Packets/Sec	
Broadcast packets sent	Indicates the total number of packets sent per second by the switch.	Packets/Sec	
Data received	Indicates the rate at which bytes were received by the port.	Mbps	
Data transmitted	Indicates the rate at which bytes of data were sent by the port.	Mbps	
Packets received	Indicates the total number of packets received by the port per second.	Packets/Sec	
Packets sent	Indicates the total number of packets sent by the port per second.	Packets/Sec	

3.3 The TCP Layer

The tests mapped to this layer measure the TCP connectivity of the Hyper-V host.

Figure 3.5: The Tcp layer

Since both the tests mapped to this layer have already been discussed in the *Monitoring Gneric Servers* document, let us proceed to take a look at the next layer.

3.4 The Application Processes Layer

The *WindowsProcesses* layer monitors the processes critical to the smooth functioning of the Hyer-V server, and also measures the resource footprint of these key processes.

Figure 3.6: The tests associated with the Application Processes layer

Since these tests too have been dealt with in the *Monitoring Unix and Windows Servers* document, let us proceed to the *Windows Services* layer

3.5 The Windows Service Layer

The *Windows Service* layer captures the applications, system, and security errors/warning events logged in the Windows event logs , and also reveals whether the core Hyper-V services are currently available or not. In addition, the layer also monitors useful Hyper-V-specific event logs and captures errors related virtual machine configurations, Hyper-V clustering, Hyper-V integration and virtual machine management services, and Hyper-V worker processes.

Figure 3.7: The tests mapped to the Windows Service layer

Since most of the tests mapped to this layer have been elaborately discussed in the *Monitoring Unix and Windows Servers* document, we will only be discussing the Hyper-V event log tests in this section.

3.5.1 Hyper-V Config Admin Log Test

The *Hyper-V Config* logs help troubleshoot issues related to virtual machine configuration files. For instance, if you have a missing or corrupt virtual machine configuration file, the entries in the *Hyper-V Config* logs will shed light on it. Using the Hyper-V Config Admin Log test, you can be alerted if any error/warning event is captured by the *Hyper-V Config logs* and can view the complete details of these events without accessing the event logs for it.

Target of the test:

A Hyper-V / Hyper-V VDI server

Agent executing the test:

An internal agent

Output of the test:

One set of results for the **FILTER** configured

Parameters of the test:

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured
3. **PORT –** Refers to the port used by the EventLog Service. Here it is null.
4. **LOGTYPE** – Refers to the type of event logs to be monitored. The default value is Microsoft-Windows-Hyper-V-Config-Admin.
5. **POLICY BASED FILTER** - Using this page, administrators can configure the event sources, event IDs, and event descriptions to be monitored by this test. In order to enable administrators to easily and accurately provide this specification, this page provides the following options:

   - Manually specify the event sources, IDs, and descriptions in the **FILTER** text area, or,
   - Select a specification from the predefined filter policies listed in the **FILTER** box

   For explicit, manual specification of the filter conditions, select the **NO** option against the **POLICY BASED FILTER** field. This is the default selection. To choose from the list of pre-configured filter policies, or to create a new filter policy and then associate the same with the test, select the **YES** option against the **POLICY BASED FILTER** field.
6. **FILTER** - If the **POLICY BASED FILTER** flag is set to **NO**, then a **FILTER** text area will appear, wherein you will have to specify the event sources, event IDs, and event descriptions to be monitored. This specification should be of the following format: *{Displayname}:{event_sources_to_be_included}:{event_sources_to_be_excluded}:{event_IDs_to_be_included}:{event_IDs_to_be_excluded}:{event_descriptions_to_be_included}:{event_descriptions_to_be_excluded}*. For example, assume that the **FILTER** text area takes the value, *OS_events:all:Browse,Print:all:none:all:none*. Here:
   - *OS_events* is the display name that will appear as a descriptor of the test in the monitor UI;
   - *all* indicates that all the event sources need to be considered while monitoring. To monitor specific event sources, provide the source names as a comma-separated list. To ensure that none of the event sources are monitored, specify *none*.
   - In the same manner, you can provide a comma-separated list of event IDs that require monitoring. The *all* in our example represents that all the event IDs need to be considered while monitoring.
   - Similarly, the *none* (following *all* in our example) is indicative of the fact that none of the event IDs need to be excluded from monitoring. On the other hand, if you want to instruct the eG Enterprise system to ignore a few event IDs during monitoring, then provide the IDs as a comma-separated list. Likewise, specifying *all* makes sure that all the event IDs are excluded from monitoring.
   - The all which follows implies that all events, regardless of description, need to be included for monitoring. To exclude all events, use none. On the other hand, if you provide a comma-separated list of event descriptions, then the events with the specified descriptions will alone be monitored. Event descriptions can be of any of the following forms - desc*, or desc, or *desc*,or desc*, or desc1*desc2, etc. desc here refers to any string that forms part of the description. A leading '*' signifies any number of leading characters, while a trailing '*' signifies any number of trailing characters.

- In the same way, you can also provide a comma-separated list of event descriptions to be excluded from monitoring. Here again, the specification can be of any of the following forms: desc*, or desc, or *desc* ,or desc* , or desc1*desc2 , etc. desc here refers to any string that forms part of the description. A leading '*' signifies any number of leading characters, while a trailing '*' signifies any number of trailing characters. In our example however, none is specified, indicating that no event descriptions are to be excluded from monitoring. If you use all instead, it would mean that all event descriptions are to be excluded from monitoring.

By default, the **FILTER** parameter contains the value*: all:all:none:all:none:all:none*. Multiple filters are to be separated by semi-colons (;).

Note:
The event sources and event IDs specified here should be exactly the same as that which appears in the Event Viewer window.

On the other hand, if the **POLICY BASED FILTER** flag is set to **YES**, then a **FILTER** list box will appear, displaying the filter policies that pre-exist in the eG Enterprise system. A filter policy typically comprises of a specific set of event sources, event IDs, and event descriptions to be monitored. This specification is built into the policy in the following format:

{Policyname}:{event_sources_to_be_included}:{event_sources_to_be_excluded}:{event_IDs_to_be_included}:{event_IDs_to_be_excluded}:{event_descriptions_to_be_included}:{event_descriptions_to_be_excluded}

To monitor a specific combination of event sources, event IDs, and event descriptions, you can choose the corresponding filter policy from the **FILTER** list box. Multiple filter policies can be so selected. Alternatively, you can modify any of the existing policies to suit your needs, or create a new filter policy. To facilitate this, a **Click here** link appears just above the test configuration section, once the **YES** option is chosen against **POLICY BASED FILTER**. Clicking on the **Click here** link leads you to a page where you can modify the existing policies or create a new one. The changed policy or the new policy can then be associated with the test by selecting the policy name from the **FILTER** list box in this page.

7. **USEWMI** - The eG agent can either use WMI to extract event log statistics or directly parse the event logs using event log APIs. If the **USEWMI** flag is **YES**, then WMI is used. If not, the event log APIs are used. This option is provided because on some Windows NT/2000 systems (especially ones with service pack 3 or lower), the use of WMI access to event logs can cause the CPU usage of the WinMgmt process to shoot up. On such systems, set the **USEWMI** parameter value to **NO**. **On the other hand, when monitoring systems that are operating on any other flavor of Windows (say, Windows 2003/XP/2008/7/Vista/12), the USEWMI flag should always be set to 'Yes'.**
8. **STATELESS ALERTS** - Typically, the eG manager generates email alerts only when the state of a specific measurement changes. A state change typically occurs only when the threshold of a measure is violated a configured number of times within a specified time window. While this ensured that the eG manager raised alarms only when the problem was severe enough, in some cases, it may cause one/more problems to go unnoticed, just because they did not result in a state change. For example,

take the case of the EventLog test. When this test captures an error event for the very first time, the eG manager will send out a **CRITICAL** email alert with the details of the error event to configured recipients. Now, the next time the test runs, if a different error event is captured, the eG manager will keep the state of the measure as **CRITICAL**, but will not send out the details of this error event to the user; thus, the second issue will remain hidden from the user. To make sure that administrators do not miss/overlook critical issues, the eG Enterprise monitoring solution provides the **stateless alerting** capability. To enable this capability for this test, set the **STATELESS ALERTS** flag to **Yes**. This will ensure that email alerts are generated for this test, regardless of whether or not the state of the measures reported by this test changes.

9. **EVENTS DURING RESTART** - By default, the **EVENTS DURING RESTART** flag is set to **Yes**. This ensures that whenever the agent is stopped and later started, the events that might have occurred during the period of non-availability of the agent are included in the number of events reported by the agent. Setting the flag to **No** ensures that the agent, when restarted, ignores the events that occurred during the time it was not available.
10. **DDFORINFORMATION** – eG Enterprise also provides you with options to restrict the amount of storage required for event log tests. Towards this end, the **DDFORINFORMATION** and **DDFORWARNING** flags have been made available in this page. By default, both these flags are set to **YES**, indicating that by default, the test generates detailed diagnostic measures for information events and warning events. If you do not want the test to generate and store detailed measures for information events, set the **DDFORINFORMATION** flag to **No**.
11. **DDFORWARNING** – To ensure that the test does not generate and store detailed measures for warning events, set the **DDFORWARNING** flag to **NO**.
12. **DD FREQUENCY** - Refers to the frequency with which detailed diagnosis measures are to be generated for this test. The default is *1:1*. This indicates that, by default, detailed measures will be generated every time this test runs, and also every time the test detects a problem. You can modify this frequency, if you so desire. Also, if you intend to disable the detailed diagnosis capability for this test, you can do so by specifying *none* against **DD FREQUENCY**.
13. **DETAILED DIAGNOSIS** - To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option. The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:
    - The eG manager license should allow the detailed diagnosis capability
    - Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
Error messages	This refers to the number of error events that were generated.	Number	A very low value (zero) indicates that no problems exist in the virtual machine configuration files. An increasing trend or high value indicates the existence of problems. Please check the *Hyper-V COnfig* Logs in the Event Log Viewer for more details.
Information messages	This refers to the number of information events generated when the test was last executed.	Number	A change in the value of this measure may indicate infrequent but successful operations performed. Please check the *Hyper-V Config* Logs in the Event Log Viewer for more details.
Warnings	This refers to the number of warnings that were generated when the test was last executed.	Number	A high value of this measure indicates v problems with virtual machine configuration files that may not have an immediate impact, but may cause future problems. Please check the *Hyper-V Config* Logs in the Event Log Viewer for more details.
Critical messages	Indicates the number of critical events that were generated when the test was last executed.	Number	A critical event is one that a virtual machine configuration file cannot automatically recover from. This measure is applicable only for Windows 2008/Windows Vista/Windows 7 systems. A very low value (zero) indicates good health. An increasing trend or high value indicates the existence of fatal/irrepairable problems.

Measurement	Description	Measurement Unit	Interpretation
			The detailed diagnosis of this measure describes all the critical events that were generated during the last measurement period. Please check the *Hyper-V Config logs* in the Event Log Viewer for more details.
Verbose messages	Indicates the number of verbose events that were generated when the test was last executed.	Number	Verbose logging provides more details in the log entry, which will enable you to troubleshoot issues better. This measure is applicable only for Windows 2008/Windows Vista/Windows 7 systems. The detailed diagnosis of this measure describes all the verbose events that were generated during the last measurement period. Please check the *Hyper-V Config Logs* in the Event Log Viewer for more details.

3.5.2 Hyper-V High Availability Admin Log Test

The *Hyper-V High Availability* logs shed light on the actions and changes that take place because of Hyper-V clustering. Using the **Hyper-V High Availability Admin Log** test, you can be alerted if any error/warning event is captured by the *Hyper-V High Availability logs* and can view the complete details of these events without accessing the event logs for it.

Target of the test:

A Hyper-V / Hyper-V VDI server

Agent executing the test:

An internal agent

Output of the test:

One set of results for the **FILTER** configured

Parameters of the test:

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured
3. **PORT –** Refers to the port used by the EventLog Service. Here it is null.
4. **LOGTYPE** – Refers to the type of event logs to be monitored. The default value is Microsoft-Windows-Hyper-V-High-Availability-Admin.
5. **POLICY BASED FILTER** - Using this page, administrators can configure the event sources, event IDs, and event descriptions to be monitored by this test. In order to enable administrators to easily and accurately provide this specification, this page provides the following options:

   - Manually specify the event sources, IDs, and descriptions in the **FILTER** text area, or,
   - Select a specification from the predefined filter policies listed in the **FILTER** box

   For explicit, manual specification of the filter conditions, select the **NO** option against the **POLICY BASED FILTER** field. This is the default selection. To choose from the list of pre-configured filter policies, or to create a new filter policy and then associate the same with the test, select the **YES** option against the **POLICY BASED FILTER** field.
6. **FILTER** - If the **POLICY BASED FILTER** flag is set to **NO**, then a **FILTER** text area will appear, wherein you will have to specify the event sources, event IDs, and event descriptions to be monitored. This specification should be of the following format: *{Displayname}:{event_sources_to_be_included}:{event_sources_to_be_excluded}:{event_IDs_to_be_included}:{event_IDs_to_be_excluded}:{event_descriptions_to_be_included}:{event_descriptions_to_be_excluded}*. For example, assume that the **FILTER** text area takes the value, *OS_events:all:Browse,Print:all:none:all:none*. Here:
   - *OS_events* is the display name that will appear as a descriptor of the test in the monitor UI;
   - *all* indicates that all the event sources need to be considered while monitoring. To monitor specific event sources, provide the source names as a comma-separated list. To ensure that none of the event sources are monitored, specify *none*.
   - In the same manner, you can provide a comma-separated list of event IDs that require monitoring. The *all* in our example represents that all the event IDs need to be considered while monitoring.
   - Similarly, the *none* (following *all* in our example) is indicative of the fact that none of the event IDs need to be excluded from monitoring. On the other hand, if you want to instruct the eG Enterprise

system to ignore a few event IDs during monitoring, then provide the IDs as a comma-separated list. Likewise, specifying *all* makes sure that all the event IDs are excluded from monitoring.

- The all which follows implies that all events, regardless of description, need to be included for monitoring. To exclude all events, use none. On the other hand, if you provide a comma-separated list of event descriptions, then the events with the specified descriptions will alone be monitored. Event descriptions can be of any of the following forms - desc*, or desc, or *desc*,or desc*, or desc1*desc2, etc. desc here refers to any string that forms part of the description. A leading '*' signifies any number of leading characters, while a trailing '*' signifies any number of trailing characters.
- In the same way, you can also provide a comma-separated list of event descriptions to be excluded from monitoring. Here again, the specification can be of any of the following forms: desc*, or desc, or *desc*,or desc*, or desc1*desc2, etc. desc here refers to any string that forms part of the description. A leading '*' signifies any number of leading characters, while a trailing '*' signifies any number of trailing characters. In our example however, none is specified, indicating that no event descriptions are to be excluded from monitoring. If you use all instead, it would mean that all event descriptions are to be excluded from monitoring.

By default, the **FILTER** parameter contains the value*: all:all:none:all:none:all:none*. Multiple filters are to be separated by semi-colons (;).

Note:
The event sources and event IDs specified here should be exactly the same as that which appears in the Event Viewer window.

On the other hand, if the **POLICY BASED FILTER** flag is set to **YES**, then a **FILTER** list box will appear, displaying the filter policies that pre-exist in the eG Enterprise system. A filter policy typically comprises of a specific set of event sources, event IDs, and event descriptions to be monitored. This specification is built into the policy in the following format:

{Policyname}:{event_sources_to_be_included}:{event_sources_to_be_excluded}:{event_IDs_to_be_included}:{event_IDs_to_be_excluded}:{event_descriptions_to_be_included}:{event_descriptions_to_be_excluded}

To monitor a specific combination of event sources, event IDs, and event descriptions, you can choose the corresponding filter policy from the **FILTER** list box. Multiple filter policies can be so selected. Alternatively, you can modify any of the existing policies to suit your needs, or create a new filter policy. To facilitate this, a **Click here** link appears just above the test configuration section, once the **YES** option is chosen against **POLICY BASED FILTER**. Clicking on the **Click here** link leads you to a page where you can modify the existing policies or create a new one. The changed policy or the new policy can then be associated with the test by selecting the policy name from the **FILTER** list box in this page.

7. **USEWMI** - The eG agent can either use WMI to extract event log statistics or directly parse the event logs using event log APIs. If the **USEWMI** flag is **YES**, then WMI is used. If not, the event log APIs are

used. This option is provided because on some Windows NT/2000 systems (especially ones with service pack 3 or lower), the use of WMI access to event logs can cause the CPU usage of the WinMgmt process to shoot up. On such systems, set the **USEWMI** parameter value to **NO**. **On the other hand, when monitoring systems that are operating on any other flavor of Windows (say, Windows 2003/XP/2008/7/Vista/12), the USEWMI flag should always be set to 'Yes'.**

8. **STATELESS ALERTS** - Typically, the eG manager generates email alerts only when the state of a specific measurement changes. A state change typically occurs only when the threshold of a measure is violated a configured number of times within a specified time window. While this ensured that the eG manager raised alarms only when the problem was severe enough, in some cases, it may cause one/more problems to go unnoticed, just because they did not result in a state change. For example, take the case of the EventLog test. When this test captures an error event for the very first time, the eG manager will send out a **CRITICAL** email alert with the details of the error event to configured recipients. Now, the next time the test runs, if a different error event is captured, the eG manager will keep the state of the measure as **CRITICAL**, but will not send out the details of this error event to the user; thus, the second issue will remain hidden from the user. To make sure that administrators do not miss/overlook critical issues, the eG Enterprise monitoring solution provides the **stateless alerting** capability. To enable this capability for this test, set the **STATELESS ALERTS** flag to **Yes**. This will ensure that email alerts are generated for this test, regardless of whether or not the state of the measures reported by this test changes.
9. **EVENTS DURING RESTART** - By default, the **EVENTS DURING RESTART** flag is set to **Yes**. This ensures that whenever the agent is stopped and later started, the events that might have occurred during the period of non-availability of the agent are included in the number of events reported by the agent. Setting the flag to **No** ensures that the agent, when restarted, ignores the events that occurred during the time it was not available.
10. **DDFORINFORMATION** – eG Enterprise also provides you with options to restrict the amount of storage required for event log tests. Towards this end, the **DDFORINFORMATION** and **DDFORWARNING** flags have been made available in this page. By default, both these flags are set to **YES**, indicating that by default, the test generates detailed diagnostic measures for information events and warning events. If you do not want the test to generate and store detailed measures for information events, set the **DDFORINFORMATION** flag to **No**.
11. **DDFORWARNING** – To ensure that the test does not generate and store detailed measures for warning events, set the **DDFORWARNING** flag to **NO**.
12. **DD FREQUENCY** - Refers to the frequency with which detailed diagnosis measures are to be generated for this test. The default is *1:1*. This indicates that, by default, detailed measures will be generated every time this test runs, and also every time the test detects a problem. You can modify this frequency, if you so desire. Also, if you intend to disable the detailed diagnosis capability for this test, you can do so by specifying *none* against **DD FREQUENCY**.
13. **DETAILED DIAGNOSIS** - To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option. The option to selectively enable/disable the detailed diagnosis

capability will be available only if the following conditions are fulfilled:

- The eG manager license should allow the detailed diagnosis capability
- Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measures of the test:

Measurement	Description	Measurement Unit	Interpretation
Error messages	This refers to the number of error events that were generated.	Number	A very low value (zero) indicates that no problems exist in Hyper- V clustering. An increasing trend or high value indicates the existence of problems. Please check the *Hyper- V High Availability* Logs in the Event Log Viewer for more details.
Information messages	This refers to the number of information events generated when the test was last executed.	Number	A change in the value of this measure may indicate infrequent but successful operations performed. Please check the *Hyper- V High Availability* Logs in the Event Log Viewer for more details.
Warning	This refers to the number of warnings that were generated when the test was last executed.	Number	A high value of this measure indicates problems with Hyper- V clustering that may not have an immediate impact, but may cause future problems. Please check the *Hyper- V High Availability* Logs in the Event Log Viewer for more details.
Critical messages	Indicates the number of critical events that were generated when the test was last executed.	Number	A critical event is one that Hyper-V cannot automatically recover from. This measure is applicable only for Windows 2008/Windows Vista/Windows 7 systems. A very low value (zero) indicates good health. An increasing trend or high value indicates the existence of fatal/irrepairable problems. The detailed diagnosis of this

Measurement	Description	Measurement Unit	Interpretation
			measure describes all the critical events that were generated during the last measurement period. Please check the *Hyper-V High Availability logs* in the Event Log Viewer for more details.
Verbose messages	Indicates the number of verbose events that were generated when the test was last executed.	Number	Verbose logging provides more details in the log entry, which will enable you to troubleshoot issues better. This measure is applicable only for Windows 2008/Windows Vista/Windows 7 systems. The detailed diagnosis of this measure describes all the verbose events that were generated during the last measurement period. Please check the *Hyper-V High Availability Logs* in the Event Log Viewer for more details.

3.5.3 Hyper-V Integration Admin Log Test

The *Hyper-V Integration* logs serve as useful source of information for analyzing errors, warnings, and general details related to the Hyper-V integration services. Using the **Hyper-V Integration Admin Log** test, you can be alerted if any error/warning event is captured by the *Hyper-V Integration logs* and can view the complete details of these events without accessing the event logs for it.

Target of the test:

A Hyper-V / Hyper-V VDI server

Agent executing the test:

An internal agent

Output of the test:

One set of results for the **FILTER** configured

Parameters of the test:

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured
3. **PORT –** Refers to the port used by the EventLog Service. Here it is null.
4. **LOGTYPE** – Refers to the type of event logs to be monitored. The default value is Microsoft-Windows-Hyper-V-Integration-Admin..
5. **POLICY BASED FILTER** - Using this page, administrators can configure the event sources, event IDs, and event descriptions to be monitored by this test. In order to enable administrators to easily and accurately provide this specification, this page provides the following options:

   - Manually specify the event sources, IDs, and descriptions in the **FILTER** text area, or,
   - Select a specification from the predefined filter policies listed in the **FILTER** box

   For explicit, manual specification of the filter conditions, select the **NO** option against the **POLICY BASED FILTER** field. This is the default selection. To choose from the list of pre-configured filter policies, or to create a new filter policy and then associate the same with the test, select the **YES** option against the **POLICY BASED FILTER** field.
6. **FILTER** - If the **POLICY BASED FILTER** flag is set to **NO**, then a **FILTER** text area will appear, wherein you will have to specify the event sources, event IDs, and event descriptions to be monitored. This specification should be of the following format: *{Displayname}:{event_sources_to_be_included}:{event_sources_to_be_excluded}:{event_IDs_to_be_included}:{event_IDs_to_be_excluded}:{event_descriptions_to_be_included}:{event_descriptions_to_be_excluded}*. For example, assume that the **FILTER** text area takes the value, *OS_events:all:Browse,Print:all:none:all:none*. Here:
   - *OS_events* is the display name that will appear as a descriptor of the test in the monitor UI;
   - *all* indicates that all the event sources need to be considered while monitoring. To monitor specific event sources, provide the source names as a comma-separated list. To ensure that none of the event sources are monitored, specify *none*.
   - In the same manner, you can provide a comma-separated list of event IDs that require monitoring. The *all* in our example represents that all the event IDs need to be considered while monitoring.
   - Similarly, the *none* (following *all* in our example) is indicative of the fact that none of the event IDs need to be excluded from monitoring. On the other hand, if you want to instruct the eG Enterprise system to ignore a few event IDs during monitoring, then provide the IDs as a comma-separated list. Likewise, specifying *all* makes sure that all the event IDs are excluded from monitoring.

- The all which follows implies that all events, regardless of description, need to be included for monitoring. To exclude all events, use none. On the other hand, if you provide a comma-separated list of event descriptions, then the events with the specified descriptions will alone be monitored. Event descriptions can be of any of the following forms - desc*, or desc, or *desc*,or desc*, or desc1*desc2, etc. desc here refers to any string that forms part of the description. A leading '*' signifies any number of leading characters, while a trailing '*' signifies any number of trailing characters.
- In the same way, you can also provide a comma-separated list of event descriptions to be excluded from monitoring. Here again, the specification can be of any of the following forms: desc*, or desc, or *desc* ,or desc* , or desc1*desc2, etc. desc here refers to any string that forms part of the description. A leading '*' signifies any number of leading characters, while a trailing '*' signifies any number of trailing characters. In our example however, none is specified, indicating that no event descriptions are to be excluded from monitoring. If you use all instead, it would mean that all event descriptions are to be excluded from monitoring.

By default, the **FILTER** parameter contains the value*: all:all:none:all:none:all:none*. Multiple filters are to be separated by semi-colons (;).

Note:
The event sources and event IDs specified here should be exactly the same as that which appears in the Event Viewer window.

On the other hand, if the **POLICY BASED FILTER** flag is set to **YES**, then a **FILTER** list box will appear, displaying the filter policies that pre-exist in the eG Enterprise system. A filter policy typically comprises of a specific set of event sources, event IDs, and event descriptions to be monitored. This specification is built into the policy in the following format:

{Policyname}:{event_sources_to_be_included}:{event_sources_to_be_excluded}:{event_IDs_to_be_included}:{event_IDs_to_be_excluded}:{event_descriptions_to_be_included}:{event_descriptions_to_be_excluded}

To monitor a specific combination of event sources, event IDs, and event descriptions, you can choose the corresponding filter policy from the **FILTER** list box. Multiple filter policies can be so selected. Alternatively, you can modify any of the existing policies to suit your needs, or create a new filter policy. To facilitate this, a **Click here** link appears just above the test configuration section, once the **YES** option is chosen against **POLICY BASED FILTER**. Clicking on the **Click here** link leads you to a page where you can modify the existing policies or create a new one. The changed policy or the new policy can then be associated with the test by selecting the policy name from the **FILTER** list box in this page.

7. **USEWMI** - The eG agent can either use WMI to extract event log statistics or directly parse the event logs using event log APIs. If the **USEWMI** flag is **YES**, then WMI is used. If not, the event log APIs are used. This option is provided because on some Windows NT/2000 systems (especially ones with service pack 3 or lower), the use of WMI access to event logs can cause the CPU usage of the WinMgmt process to shoot up. On such systems, set the **USEWMI** parameter value to **NO**. **On the**

other hand, when monitoring systems that are operating on any other flavor of Windows (say, Windows 2003/XP/2008/7/Vista/12), the USEWMI flag should always be set to 'Yes'.

8. **STATELESS ALERTS** - Typically, the eG manager generates email alerts only when the state of a specific measurement changes. A state change typically occurs only when the threshold of a measure is violated a configured number of times within a specified time window. While this ensured that the eG manager raised alarms only when the problem was severe enough, in some cases, it may cause one/more problems to go unnoticed, just because they did not result in a state change. For example, take the case of the EventLog test. When this test captures an error event for the very first time, the eG manager will send out a **CRITICAL** email alert with the details of the error event to configured recipients. Now, the next time the test runs, if a different error event is captured, the eG manager will keep the state of the measure as **CRITICAL**, but will not send out the details of this error event to the user; thus, the second issue will remain hidden from the user. To make sure that administrators do not miss/overlook critical issues, the eG Enterprise monitoring solution provides the **stateless alerting** capability. To enable this capability for this test, set the **STATELESS ALERTS** flag to **Yes**. This will ensure that email alerts are generated for this test, regardless of whether or not the state of the measures reported by this test changes.
9. **EVENTS DURING RESTART** - By default, the **EVENTS DURING RESTART** flag is set to **Yes**. This ensures that whenever the agent is stopped and later started, the events that might have occurred during the period of non-availability of the agent are included in the number of events reported by the agent. Setting the flag to **No** ensures that the agent, when restarted, ignores the events that occurred during the time it was not available.
10. **DDFORINFORMATION** – eG Enterprise also provides you with options to restrict the amount of storage required for event log tests. Towards this end, the **DDFORINFORMATION** and **DDFORWARNING** flags have been made available in this page. By default, both these flags are set to **YES**, indicating that by default, the test generates detailed diagnostic measures for information events and warning events. If you do not want the test to generate and store detailed measures for information events, set the **DDFORINFORMATION** flag to **No**.
11. **DDFORWARNING** – To ensure that the test does not generate and store detailed measures for warning events, set the **DDFORWARNING** flag to **NO**.
12. **DD FREQUENCY** - Refers to the frequency with which detailed diagnosis measures are to be generated for this test. The default is *1:1*. This indicates that, by default, detailed measures will be generated every time this test runs, and also every time the test detects a problem. You can modify this frequency, if you so desire. Also, if you intend to disable the detailed diagnosis capability for this test, you can do so by specifying *none* against **DD FREQUENCY**.
13. **DETAILED DIAGNOSIS** - To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option. The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:
    - The eG manager license should allow the detailed diagnosis capability

- Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
Error messages	This refers to the number of error events that were generated.	Number	A very low value (zero) indicates that no problems exist in integration services. An increasing trend or high value indicates the existence of problems. Please check the *Hyper-V Integration* Logs in the Event Log Viewer for more details.
Information messages	This refers to the number of information events generated when the test was last executed.	Number	A change in the value of this measure may indicate infrequent but successful operations performed. Please check the *Hyper-V Integration* Logs in the Event Log Viewer for more details.
Warnings	This refers to the number of warnings that were generated when the test was last executed.	Number	A high value of this measure indicates problems with Hyper- V integration services that may not have an immediate impact, but may cause future problems. Please check the *Hyper-V Integration* Logs in the Event Log Viewer for more details.
Critical messages	Indicates the number of critical events that were generated when the test was last executed.	Number	A critical event is one that the Hyper-V integration services cannot automatically recover from. This measure is applicable only for Windows 2008/Windows Vista/Windows 7 systems. A very low value (zero) indicates good health. An increasing trend or high value indicates the existence of fatal/irrepairable problems.

Measurement	Description	Measurement Unit	Interpretation
			The detailed diagnosis of this measure describes all the critical events that were generated during the last measurement period. Please check the *Hyper-V Integration logs* in the Event Log Viewer for more details.
Verbose messages	Indicates the number of verbose events that were generated when the test was last executed.	Number	Verbose logging provides more details in the log entry, which will enable you to troubleshoot issues better. This measure is applicable only for Windows 2008/Windows Vista/Windows 7 systems. The detailed diagnosis of this measure describes all the verbose events that were generated during the last measurement period. Please check the *Hyper-V Integration Logs* in the Event Log Viewer for more details.

3.5.4 Hyper-V VMMS Storage Log Test

The *Hyper-V VMMS logs* capture storage-related issues. Using the **Hyper-V VMMS Storage Log** test, you can be alerted if any error/warning event is captured by the *Hyper-V VMMS logs* and can view the complete details of these events without accessing the event logs for it.

Target of the test:

A Hyper-V / Hyper-V VDI server

Agent executing the test:

An internal agent

Output of the test:

One set of results for the **FILTER** configured

Parameters of the test:

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured
3. **PORT –** Refers to the port used by the EventLog Service. Here it is null.
4. **LOGTYPE** – Refers to the type of event logs to be monitored. The default value is Microsoft-Windows-Hyper-V-VMMS-Storage.
5. **POLICY BASED FILTER** - Using this page, administrators can configure the event sources, event IDs, and event descriptions to be monitored by this test. In order to enable administrators to easily and accurately provide this specification, this page provides the following options:

   - Manually specify the event sources, IDs, and descriptions in the **FILTER** text area, or,
   - Select a specification from the predefined filter policies listed in the **FILTER** box

   For explicit, manual specification of the filter conditions, select the **NO** option against the **POLICY BASED FILTER** field. This is the default selection. To choose from the list of pre-configured filter policies, or to create a new filter policy and then associate the same with the test, select the **YES** option against the **POLICY BASED FILTER** field.
6. **FILTER** - If the **POLICY BASED FILTER** flag is set to **NO**, then a **FILTER** text area will appear, wherein you will have to specify the event sources, event IDs, and event descriptions to be monitored. This specification should be of the following format: *{Displayname}:{event_sources_to_be_included}:{event_sources_to_be_excluded}:{event_IDs_to_be_included}:{event_IDs_to_be_excluded}:{event_descriptions_to_be_included}:{event_descriptions_to_be_excluded}*. For example, assume that the **FILTER** text area takes the value, *OS_events:all:Browse,Print:all:none:all:none*. Here:
   - *OS_events* is the display name that will appear as a descriptor of the test in the monitor UI;
   - *all* indicates that all the event sources need to be considered while monitoring. To monitor specific event sources, provide the source names as a comma-separated list. To ensure that none of the event sources are monitored, specify *none*.
   - In the same manner, you can provide a comma-separated list of event IDs that require monitoring. The *all* in our example represents that all the event IDs need to be considered while monitoring.
   - Similarly, the *none* (following *all* in our example) is indicative of the fact that none of the event IDs need to be excluded from monitoring. On the other hand, if you want to instruct the eG Enterprise system to ignore a few event IDs during monitoring, then provide the IDs as a comma-separated list. Likewise, specifying *all* makes sure that all the event IDs are excluded from monitoring.

- The all which follows implies that all events, regardless of description, need to be included for monitoring. To exclude all events, use none. On the other hand, if you provide a comma-separated list of event descriptions, then the events with the specified descriptions will alone be monitored. Event descriptions can be of any of the following forms - desc*, or desc, or *desc*,or desc*, or desc1*desc2, etc. desc here refers to any string that forms part of the description. A leading '*' signifies any number of leading characters, while a trailing '*' signifies any number of trailing characters.
- In the same way, you can also provide a comma-separated list of event descriptions to be excluded from monitoring. Here again, the specification can be of any of the following forms: desc*, or desc, or *desc*,or desc*, or desc1*desc2, etc. desc here refers to any string that forms part of the description. A leading '*' signifies any number of leading characters, while a trailing '*' signifies any number of trailing characters. In our example however, none is specified, indicating that no event descriptions are to be excluded from monitoring. If you use all instead, it would mean that all event descriptions are to be excluded from monitoring.

By default, the **FILTER** parameter contains the value*: all:all:none:all:none:all:none*. Multiple filters are to be separated by semi-colons (;).

Note:
The event sources and event IDs specified here should be exactly the same as that which appears in the Event Viewer window.

On the other hand, if the **POLICY BASED FILTER** flag is set to **YES**, then a **FILTER** list box will appear, displaying the filter policies that pre-exist in the eG Enterprise system. A filter policy typically comprises of a specific set of event sources, event IDs, and event descriptions to be monitored. This specification is built into the policy in the following format:

{Policyname}:{event_sources_to_be_included}:{event_sources_to_be_excluded}:{event_IDs_to_be_included}:{event_IDs_to_be_excluded}:{event_descriptions_to_be_included}:{event_descriptions_to_be_excluded}

To monitor a specific combination of event sources, event IDs, and event descriptions, you can choose the corresponding filter policy from the **FILTER** list box. Multiple filter policies can be so selected. Alternatively, you can modify any of the existing policies to suit your needs, or create a new filter policy. To facilitate this, a **Click here** link appears just above the test configuration section, once the **YES** option is chosen against **POLICY BASED FILTER**. Clicking on the **Click here** link leads you to a page where you can modify the existing policies or create a new one. The changed policy or the new policy can then be associated with the test by selecting the policy name from the **FILTER** list box in this page.

7. **USEWMI** - The eG agent can either use WMI to extract event log statistics or directly parse the event logs using event log APIs. If the **USEWMI** flag is **YES**, then WMI is used. If not, the event log APIs are used. This option is provided because on some Windows NT/2000 systems (especially ones with service pack 3 or lower), the use of WMI access to event logs can cause the CPU usage of the WinMgmt process to shoot up. On such systems, set the **USEWMI** parameter value to **NO**. **On the**

other hand, when monitoring systems that are operating on any other flavor of Windows (say, Windows 2003/XP/2008/7/Vista/12), the USEWMI flag should always be set to 'Yes'.

8. **STATELESS ALERTS** - Typically, the eG manager generates email alerts only when the state of a specific measurement changes. A state change typically occurs only when the threshold of a measure is violated a configured number of times within a specified time window. While this ensured that the eG manager raised alarms only when the problem was severe enough, in some cases, it may cause one/more problems to go unnoticed, just because they did not result in a state change. For example, take the case of the EventLog test. When this test captures an error event for the very first time, the eG manager will send out a **CRITICAL** email alert with the details of the error event to configured recipients. Now, the next time the test runs, if a different error event is captured, the eG manager will keep the state of the measure as **CRITICAL**, but will not send out the details of this error event to the user; thus, the second issue will remain hidden from the user. To make sure that administrators do not miss/overlook critical issues, the eG Enterprise monitoring solution provides the **stateless alerting** capability. To enable this capability for this test, set the **STATELESS ALERTS** flag to **Yes**. This will ensure that email alerts are generated for this test, regardless of whether or not the state of the measures reported by this test changes.
9. **EVENTS DURING RESTART** - By default, the **EVENTS DURING RESTART** flag is set to **Yes**. This ensures that whenever the agent is stopped and later started, the events that might have occurred during the period of non-availability of the agent are included in the number of events reported by the agent. Setting the flag to **No** ensures that the agent, when restarted, ignores the events that occurred during the time it was not available.
10. **DDFORINFORMATION** – eG Enterprise also provides you with options to restrict the amount of storage required for event log tests. Towards this end, the **DDFORINFORMATION** and **DDFORWARNING** flags have been made available in this page. By default, both these flags are set to **YES**, indicating that by default, the test generates detailed diagnostic measures for information events and warning events. If you do not want the test to generate and store detailed measures for information events, set the **DDFORINFORMATION** flag to **No**.
11. **DDFORWARNING** – To ensure that the test does not generate and store detailed measures for warning events, set the **DDFORWARNING** flag to **NO**.
12. **DD FREQUENCY** - Refers to the frequency with which detailed diagnosis measures are to be generated for this test. The default is *1:1*. This indicates that, by default, detailed measures will be generated every time this test runs, and also every time the test detects a problem. You can modify this frequency, if you so desire. Also, if you intend to disable the detailed diagnosis capability for this test, you can do so by specifying *none* against **DD FREQUENCY**.
13. **DETAILED DIAGNOSIS** - To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option. The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:
    - The eG manager license should allow the detailed diagnosis capability

- Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
Error messages	This refers to the number of error events that were generated.	Number	A very low value (zero) indicates that no problems exist in the storage. An increasing trend or high value indicates the existence of problems. Please check the *Hyper-V VMMS* Logs in the Event Log Viewer for more details.
Information messages	This refers to the number of information events generated when the test was last executed.	Number	A change in the value of this measure may indicate infrequent but successful operations performed. Please check the *Hyper-V VMMS* Logs in the Event Log Viewer for more details.
Warnings	This refers to the number of warnings that were generated when the test was last executed.	Number	A high value of this measure indicates problems with storage that may not have an immediate impact, but may cause future problems. Please check the *Hyper-V VMMS* Logs in the Event Log Viewer for more details.
Critical messages	Indicates the number of critical events that were generated when the test was last executed.	Number	A critical event is one that the storage cannot automatically recover from. This measure is applicable only for Windows 2008/Windows Vista/Windows 7 systems. A very low value (zero) indicates good health. An increasing trend or high value indicates the existence of fatal/irrepairable problems. The detailed diagnosis of this

Measurement	Description	Measurement Unit	Interpretation
			measure describes all the critical events that were generated during the last measurement period. Please check the *Hyper-V VMMS logs* in the Event Log Viewer for more details.
Verbose messages	Indicates the number of verbose events that were generated when the test was last executed.	Number	Verbose logging provides more details in the log entry, which will enable you to troubleshoot issues better. This measure is applicable only for Windows 2008/Windows Vista/Windows 7 systems. The detailed diagnosis of this measure describes all the verbose events that were generated during the last measurement period. Please check the *Hyper-V VMMS logs* in the Event Log Viewer for more details.

3.5.5 Hyper-V VMMS Admin Log Test

The *Hyper-V VMMS logs* capture events related to the virtual machine management services. Using the **Hyper-V VMMS Admin Log** test, you can be alerted if any error/warning event is captured by the *Hyper-V VMMS logs* and can view the complete details of these events without accessing the event logs for it.

Target of the test:

A Hyper-V / Hyper-V VDI server

Agent executing the test:

An internal agent

Output of the test:

One set of results for the **FILTER** configured

Parameters of the test:

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured
3. **PORT –** Refers to the port used by the EventLog Service. Here it is null.
4. **LOGTYPE** – Refers to the type of event logs to be monitored. The default value is Microsoft-Windows-Hyper-V-VMMS-Admin.
5. **POLICY BASED FILTER** - Using this page, administrators can configure the event sources, event IDs, and event descriptions to be monitored by this test. In order to enable administrators to easily and accurately provide this specification, this page provides the following options:

   - Manually specify the event sources, IDs, and descriptions in the **FILTER** text area, or,
   - Select a specification from the predefined filter policies listed in the **FILTER** box

   For explicit, manual specification of the filter conditions, select the **NO** option against the **POLICY BASED FILTER** field. This is the default selection. To choose from the list of pre-configured filter policies, or to create a new filter policy and then associate the same with the test, select the **YES** option against the **POLICY BASED FILTER** field.
6. **FILTER** - If the **POLICY BASED FILTER** flag is set to **NO**, then a **FILTER** text area will appear, wherein you will have to specify the event sources, event IDs, and event descriptions to be monitored. This specification should be of the following format: *{Displayname}:{event_sources_to_be_included}:{event_sources_to_be_excluded}:{event_IDs_to_be_included}:{event_IDs_to_be_excluded}:{event_descriptions_to_be_included}:{event_descriptions_to_be_excluded}*. For example, assume that the **FILTER** text area takes the value, *OS_events:all:Browse,Print:all:none:all:none*. Here:
   - *OS_events* is the display name that will appear as a descriptor of the test in the monitor UI;
   - *all* indicates that all the event sources need to be considered while monitoring. To monitor specific event sources, provide the source names as a comma-separated list. To ensure that none of the event sources are monitored, specify *none*.
   - In the same manner, you can provide a comma-separated list of event IDs that require monitoring. The *all* in our example represents that all the event IDs need to be considered while monitoring.
   - Similarly, the *none* (following *all* in our example) is indicative of the fact that none of the event IDs need to be excluded from monitoring. On the other hand, if you want to instruct the eG Enterprise system to ignore a few event IDs during monitoring, then provide the IDs as a comma-separated list. Likewise, specifying *all* makes sure that all the event IDs are excluded from monitoring.
   - The all which follows implies that all events, regardless of description, need to be included for

monitoring. To exclude all events, use none. On the other hand, if you provide a comma-separated list of event descriptions, then the events with the specified descriptions will alone be monitored. Event descriptions can be of any of the following forms - desc*, or desc, or *desc*,or desc*, or desc1*desc2, etc. desc here refers to any string that forms part of the description. A leading '*' signifies any number of leading characters, while a trailing '*' signifies any number of trailing characters.

- In the same way, you can also provide a comma-separated list of event descriptions to be excluded from monitoring. Here again, the specification can be of any of the following forms: desc*, or desc, or *desc*,or desc*, or desc1*desc2, etc. desc here refers to any string that forms part of the description. A leading '*' signifies any number of leading characters, while a trailing '*' signifies any number of trailing characters. In our example however, none is specified, indicating that no event descriptions are to be excluded from monitoring. If you use all instead, it would mean that all event descriptions are to be excluded from monitoring.

By default, the **FILTER** parameter contains the value*: all:all:none:all:none:all:none*. Multiple filters are to be separated by semi-colons (;).

Note:
The event sources and event IDs specified here should be exactly the same as that which appears in the Event Viewer window.

On the other hand, if the **POLICY BASED FILTER** flag is set to **YES**, then a **FILTER** list box will appear, displaying the filter policies that pre-exist in the eG Enterprise system. A filter policy typically comprises of a specific set of event sources, event IDs, and event descriptions to be monitored. This specification is built into the policy in the following format:

{Policyname}:{event_sources_to_be_included}:{event_sources_to_be_excluded}:{event_IDs_to_be_included}:{event_IDs_to_be_excluded}:{event_descriptions_to_be_included}:{event_descriptions_to_be_excluded}

To monitor a specific combination of event sources, event IDs, and event descriptions, you can choose the corresponding filter policy from the **FILTER** list box. Multiple filter policies can be so selected. Alternatively, you can modify any of the existing policies to suit your needs, or create a new filter policy. To facilitate this, a **Click here** link appears just above the test configuration section, once the **YES** option is chosen against **POLICY BASED FILTER**. Clicking on the **Click here** link leads you to a page where you can modify the existing policies or create a new one. The changed policy or the new policy can then be associated with the test by selecting the policy name from the **FILTER** list box in this page.

7. **USEWMI** - The eG agent can either use WMI to extract event log statistics or directly parse the event logs using event log APIs. If the **USEWMI** flag is **YES**, then WMI is used. If not, the event log APIs are used. This option is provided because on some Windows NT/2000 systems (especially ones with service pack 3 or lower), the use of WMI access to event logs can cause the CPU usage of the WinMgmt process to shoot up. On such systems, set the **USEWMI** parameter value to **NO**. **On the other hand, when monitoring systems that are operating on any other flavor of Windows (say,**

Windows 2003/XP/2008/7/Vista/12), the USEWMI flag should always be set to 'Yes'.

8. **STATELESS ALERTS** - Typically, the eG manager generates email alerts only when the state of a specific measurement changes. A state change typically occurs only when the threshold of a measure is violated a configured number of times within a specified time window. While this ensured that the eG manager raised alarms only when the problem was severe enough, in some cases, it may cause one/more problems to go unnoticed, just because they did not result in a state change. For example, take the case of the EventLog test. When this test captures an error event for the very first time, the eG manager will send out a **CRITICAL** email alert with the details of the error event to configured recipients. Now, the next time the test runs, if a different error event is captured, the eG manager will keep the state of the measure as **CRITICAL**, but will not send out the details of this error event to the user; thus, the second issue will remain hidden from the user. To make sure that administrators do not miss/overlook critical issues, the eG Enterprise monitoring solution provides the **stateless alerting** capability. To enable this capability for this test, set the **STATELESS ALERTS** flag to **Yes**. This will ensure that email alerts are generated for this test, regardless of whether or not the state of the measures reported by this test changes.
9. **EVENTS DURING RESTART** - By default, the **EVENTS DURING RESTART** flag is set to **Yes**. This ensures that whenever the agent is stopped and later started, the events that might have occurred during the period of non-availability of the agent are included in the number of events reported by the agent. Setting the flag to **No** ensures that the agent, when restarted, ignores the events that occurred during the time it was not available.
10. **DDFORINFORMATION** – eG Enterprise also provides you with options to restrict the amount of storage required for event log tests. Towards this end, the **DDFORINFORMATION** and **DDFORWARNING** flags have been made available in this page. By default, both these flags are set to **YES**, indicating that by default, the test generates detailed diagnostic measures for information events and warning events. If you do not want the test to generate and store detailed measures for information events, set the **DDFORINFORMATION** flag to **No**.
11. **DDFORWARNING** – To ensure that the test does not generate and store detailed measures for warning events, set the **DDFORWARNING** flag to **NO**.
12. **DD FREQUENCY** - Refers to the frequency with which detailed diagnosis measures are to be generated for this test. The default is *1:1*. This indicates that, by default, detailed measures will be generated every time this test runs, and also every time the test detects a problem. You can modify this frequency, if you so desire. Also, if you intend to disable the detailed diagnosis capability for this test, you can do so by specifying *none* against **DD FREQUENCY**.
13. **DETAILED DIAGNOSIS** - To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option. The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:
    - The eG manager license should allow the detailed diagnosis capability
    - Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
Error messages	This refers to the number of error events that were generated.	Number	A very low value (zero) indicates that no problems exist in the virtual machine management services. An increasing trend or high value indicates the existence of problems. Please check the *Hyper-V VMMS* Logs in the Event Log Viewer for more details.
Information messages	This refers to the number of information events generated when the test was last executed.	Number	A change in the value of this measure may indicate infrequent but successful operations performed. Please check the *Hyper-V VMMS* Logs in the Event Log Viewer for more details.
Warnings	This refers to the number of warnings that were generated when the test was last executed.	Number	A high value of this measure indicates problems with Hyper-V VMMS that may not have an immediate impact, but may cause future problems. Please check the *Hyper-V VMMS* Logs in the Event Log Viewer for more details.
Critical messages	Indicates the number of critical events that were generated when the test was last executed.	Number	A critical event is one that the Hyper- V VMMS cannot automatically recover from. This measure is applicable only for Windows 2008/Windows Vista/Windows 7 systems.

Measurement	Description	Measurement Unit	Interpretation
			A very low value (zero) indicates good health. An increasing trend or high value indicates the existence of fatal/irrepairable problems. The detailed diagnosis of this measure describes all the critical events that were generated during the last measurement period. Please check the *Hyper-V VMMS logs* in the Event Log Viewer for more details.
Verbose messages	Indicates the number of verbose events that were generated when the test was last executed.	Number	Verbose logging provides more details in the log entry, which will enable you to troubleshoot issues better. This measure is applicable only for Windows 2008/Windows Vista/Windows 7 systems. The detailed diagnosis of this measure describes all the verbose events that were generated during the last measurement period. Please check the *Hyper-V VMMS logs* in the Event Log Viewer for more details.

3.5.6 Hyper-V Worker Admin Log Test

The *Hyper-V Worker logs* file events related to the worker processes used for the actual running of the virtual machines. Using the **Hyper-V Worker Admin Log** test, you can be alerted if any error/warning event is captured by the *Hyper-V Worker logs* and can view the complete details of these events without accessing the event logs for it.

Target of the test:

A Hyper-V / Hyper-V VDI server

Agent executing the test:

An internal agent

Output of the test:

One set of results for the **FILTER** configured

Parameters of the test:

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured
3. **PORT –** Refers to the port used by the EventLog Service. Here it is null.
4. **LOGTYPE** – Refers to the type of event logs to be monitored. The default value is Microsoft-Windows-Hyper-V-Worker-Admin.
5. **POLICY BASED FILTER** - Using this page, administrators can configure the event sources, event IDs, and event descriptions to be monitored by this test. In order to enable administrators to easily and accurately provide this specification, this page provides the following options:

   - Manually specify the event sources, IDs, and descriptions in the **FILTER** text area, or,
   - Select a specification from the predefined filter policies listed in the **FILTER** box

   For explicit, manual specification of the filter conditions, select the **NO** option against the **POLICY BASED FILTER** field. This is the default selection. To choose from the list of pre-configured filter policies, or to create a new filter policy and then associate the same with the test, select the **YES** option against the **POLICY BASED FILTER** field.
6. **FILTER** - If the **POLICY BASED FILTER** flag is set to **NO**, then a **FILTER** text area will appear, wherein you will have to specify the event sources, event IDs, and event descriptions to be monitored. This specification should be of the following format: *{Displayname}:{event_sources_to_be_included}:{event_sources_to_be_excluded}:{event_IDs_to_be_included}:{event_IDs_to_be_excluded}:{event_descriptions_to_be_included}:{event_descriptions_to_be_excluded}*. For example, assume that the **FILTER** text area takes the value, *OS_events:all:Browse,Print:all:none:all:none*. Here:
   - *OS_events* is the display name that will appear as a descriptor of the test in the monitor UI;
   - *all* indicates that all the event sources need to be considered while monitoring. To monitor specific event sources, provide the source names as a comma-separated list. To ensure that none of the

event sources are monitored, specify *none*.

- In the same manner, you can provide a comma-separated list of event IDs that require monitoring. The *all* in our example represents that all the event IDs need to be considered while monitoring.
- Similarly, the *none* (following *all* in our example) is indicative of the fact that none of the event IDs need to be excluded from monitoring. On the other hand, if you want to instruct the eG Enterprise system to ignore a few event IDs during monitoring, then provide the IDs as a comma-separated list. Likewise, specifying *all* makes sure that all the event IDs are excluded from monitoring.
- The all which follows implies that all events, regardless of description, need to be included for monitoring. To exclude all events, use none. On the other hand, if you provide a comma-separated list of event descriptions, then the events with the specified descriptions will alone be monitored. Event descriptions can be of any of the following forms - desc*, or desc, or *desc*,or desc*, or desc1*desc2, etc. desc here refers to any string that forms part of the description. A leading '*' signifies any number of leading characters, while a trailing '*' signifies any number of trailing characters.
- In the same way, you can also provide a comma-separated list of event descriptions to be excluded from monitoring. Here again, the specification can be of any of the following forms: desc*, or desc, or *desc* ,or desc* , or desc1*desc2, etc. desc here refers to any string that forms part of the description. A leading '*' signifies any number of leading characters, while a trailing '*' signifies any number of trailing characters. In our example however, none is specified, indicating that no event descriptions are to be excluded from monitoring. If you use all instead, it would mean that all event descriptions are to be excluded from monitoring.

By default, the **FILTER** parameter contains the value*: all:all:none:all:none:all:none*. Multiple filters are to be separated by semi-colons (;).

Note:
The event sources and event IDs specified here should be exactly the same as that which appears in the Event Viewer window.

On the other hand, if the **POLICY BASED FILTER** flag is set to **YES**, then a **FILTER** list box will appear, displaying the filter policies that pre-exist in the eG Enterprise system. A filter policy typically comprises of a specific set of event sources, event IDs, and event descriptions to be monitored. This specification is built into the policy in the following format:

{Policyname}:{event_sources_to_be_included}:{event_sources_to_be_excluded}:{event_IDs_to_be_included}:{event_IDs_to_be_excluded}:{event_descriptions_to_be_included}:{event_descriptions_to_be_excluded}

To monitor a specific combination of event sources, event IDs, and event descriptions, you can choose the corresponding filter policy from the **FILTER** list box. Multiple filter policies can be so selected. Alternatively, you can modify any of the existing policies to suit your needs, or create a new filter policy. To facilitate this, a **Click here** link appears just above the test configuration section, once the **YES** option is chosen against **POLICY BASED FILTER**. Clicking on the **Click here** link leads you to a page where you can modify the existing policies or create a new one. The changed policy or the new policy can then be associated with the test by selecting the policy name from the **FILTER** list box in this

page.

7. **USEWMI** - The eG agent can either use WMI to extract event log statistics or directly parse the event logs using event log APIs. If the **USEWMI** flag is **YES**, then WMI is used. If not, the event log APIs are used. This option is provided because on some Windows NT/2000 systems (especially ones with service pack 3 or lower), the use of WMI access to event logs can cause the CPU usage of the WinMgmt process to shoot up. On such systems, set the **USEWMI** parameter value to **NO**. **On the other hand, when monitoring systems that are operating on any other flavor of Windows (say, Windows 2003/XP/2008/7/Vista/12), the USEWMI flag should always be set to 'Yes'.**
8. **STATELESS ALERTS** - Typically, the eG manager generates email alerts only when the state of a specific measurement changes. A state change typically occurs only when the threshold of a measure is violated a configured number of times within a specified time window. While this ensured that the eG manager raised alarms only when the problem was severe enough, in some cases, it may cause one/more problems to go unnoticed, just because they did not result in a state change. For example, take the case of the EventLog test. When this test captures an error event for the very first time, the eG manager will send out a **CRITICAL** email alert with the details of the error event to configured recipients. Now, the next time the test runs, if a different error event is captured, the eG manager will keep the state of the measure as **CRITICAL**, but will not send out the details of this error event to the user; thus, the second issue will remain hidden from the user. To make sure that administrators do not miss/overlook critical issues, the eG Enterprise monitoring solution provides the **stateless alerting** capability. To enable this capability for this test, set the **STATELESS ALERTS** flag to **Yes**. This will ensure that email alerts are generated for this test, regardless of whether or not the state of the measures reported by this test changes.
9. **EVENTS DURING RESTART** - By default, the **EVENTS DURING RESTART** flag is set to **Yes**. This ensures that whenever the agent is stopped and later started, the events that might have occurred during the period of non-availability of the agent are included in the number of events reported by the agent. Setting the flag to **No** ensures that the agent, when restarted, ignores the events that occurred during the time it was not available.
10. **DDFORINFORMATION** – eG Enterprise also provides you with options to restrict the amount of storage required for event log tests. Towards this end, the **DDFORINFORMATION** and **DDFORWARNING** flags have been made available in this page. By default, both these flags are set to **YES**, indicating that by default, the test generates detailed diagnostic measures for information events and warning events. If you do not want the test to generate and store detailed measures for information events, set the **DDFORINFORMATION** flag to **No**.
11. **DDFORWARNING** – To ensure that the test does not generate and store detailed measures for warning events, set the **DDFORWARNING** flag to **NO**.
12. **DD FREQUENCY** - Refers to the frequency with which detailed diagnosis measures are to be generated for this test. The default is *1:1*. This indicates that, by default, detailed measures will be generated every time this test runs, and also every time the test detects a problem. You can modify this frequency, if you so desire. Also, if you intend to disable the detailed diagnosis capability for this test, you can do so by specifying *none* against **DD FREQUENCY**.

13. **DETAILED DIAGNOSIS** - To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option. The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:
    - The eG manager license should allow the detailed diagnosis capability
    - Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
Error messages	This refers to the number of error events that were generated.	Number	A very low value (zero) indicates that no problems exist in the worker processes. An increasing trend or high value indicates the existence of problems. Please check the *Hyper-V Worker* Logs in the Event Log Viewer for more details.
Information messages	This refers to the number of information events generated when the test was last executed.	Number	A change in the value of this measure may indicate infrequent but successful operations performed. Please check the *Hyper-V Worker* Logs in the Event Log Viewer for more details.
Warnings	This refers to the number of warnings that were generated when the test was last executed.	Number	A high value of this measure indicates problems with worker processes that may not have an immediate impact, but may cause future problems. Please check the *Hyper-V Worker* Logs in the Event Log Viewer for more details.
Critical messages	Indicates the number of critical events that were generated when the test was last executed.	Number	A critical event is one that the worker processes cannot automatically recover from. This measure is applicable only for Windows 2008/Windows Vista/Windows 7 systems. A very low value (zero) indicates good health. An increasing trend or high value indicates the existence of fatal/irrepairable problems.

Measurement	Description	Measurement Unit	Interpretation
			The detailed diagnosis of this measure describes all the critical events that were generated during the last measurement period. Please check the *Hyper-V Worker logs* in the Event Log Viewer for more details.
Verbose messages	Indicates the number of verbose events that were generated when the test was last executed.	Number	Verbose logging provides more details in the log entry, which will enable you to troubleshoot issues better. This measure is applicable only for Windows 2008/Windows Vista/Windows 7 systems. The detailed diagnosis of this measure describes all the verbose events that were generated during the last measurement period. Please check the *Hyper-V Worker logs* in the Event Log Viewer for more details.

3.6 Outside View of VMs Layer

To be able to accurately assess the physical resource usage on the Hyper-V server, and to precisely identify the root-cause for any sudden/consistent resource drains on the server, the knowledge of the resource utilization of the host operating system and the root partition alone might not suffice. Administrators also need to know how each VM on the server uses the available physical resources, so that resource-intensive VMs can be promptly isolated.

Using the tests associated with the **Outside View of VMs** layer reports the powered-on status of every VM and also reveals the relative resource usage of the VMs, thereby pointing administrators to the source of a physical resource contention on the server – is it the host operating system or is it one/more of the VMs?

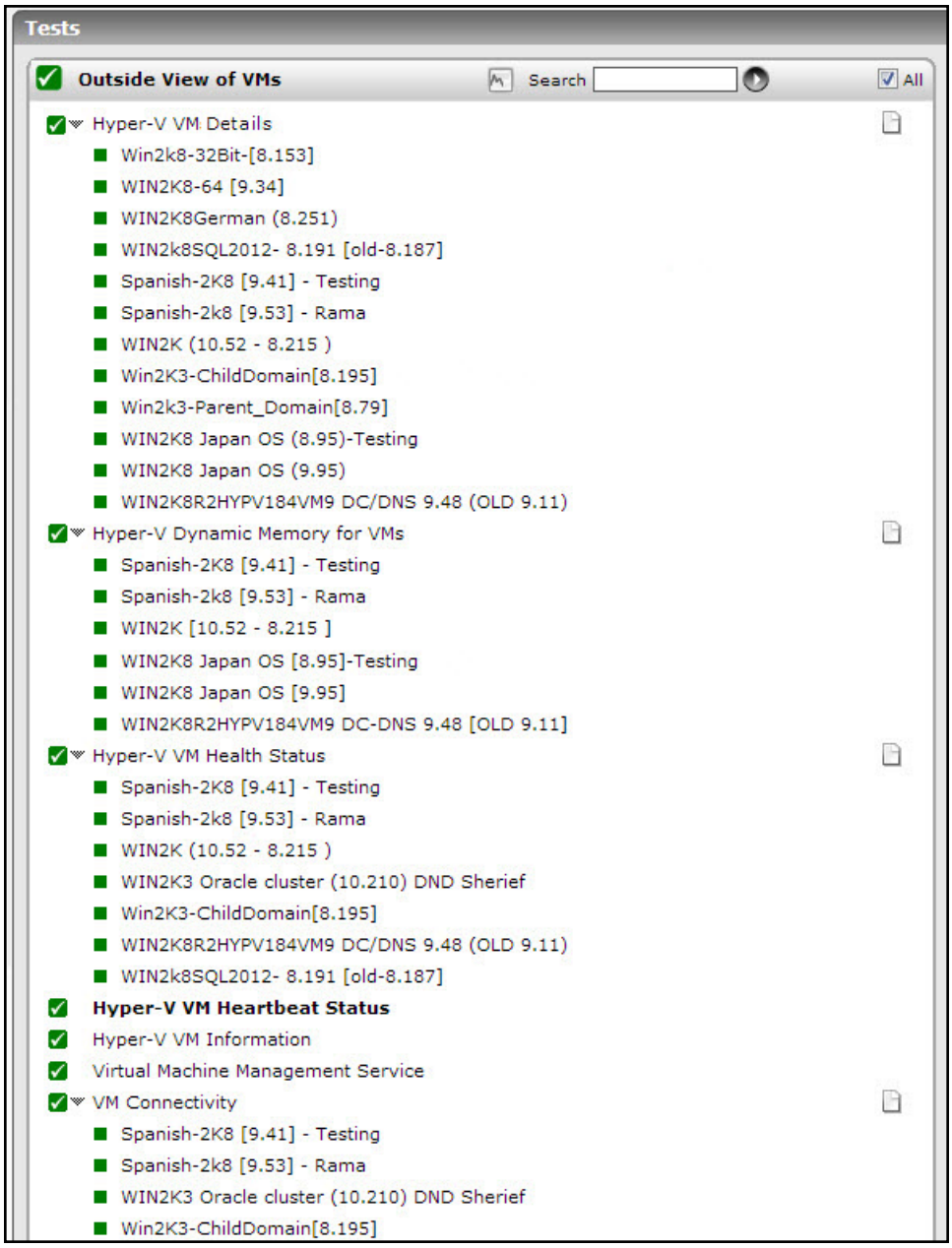


Figure 3.8: Figure 2.8: The tests linked to the Outside View of VMs layer

3.6.1 Hyper-V VM Details Test

This test monitors the amount of the physical server's resources that each guest on a Hyper-V server is taking up. Using the metrics reported by this test, administrators can determine which virtual guest is taking up most CPU, which guest is generating the most network traffic, which guest is taking up the maximum memory utilization, which guest has the maximum disk activity, etc.

Target o the test:

A Hyper-V / Hyper-V VDI server

Agent executing the test:

An internal agent

Output of the test:

One set of results for every guest operating on the monitored Hyper-V server

Parameters of the test:

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured.
3. **PORT** - The port at which the **HOST** listens. By default, this is NULL.
4. **INSIDE VIEW USING** - By default, this test communicates with every VM remotely and extracts "inside view" metrics. Therefore, by default, the **INSIDE VIEW USING** flag is set to **Remote connection to VM (Windows)**.

   Typically, to establish this remote connection, eG Enterprise requires that the eG agent be configured with domain administrator privileges. In high-security environments, where the IT staff might have reservations about exposing the credentials of their domain administrators, this approach to extracting "inside view" metrics might not be preferred. In such environments therefore, eG Enterprise provides administrators the option to deploy a piece of software called the **eG VM Agent** on every Windows VM; this VM agent allows the eG agent to collect "inside view" metrics from the Windows VMs **without domain administrator rights.** Refer to Section 1.3.2 for more details on the **eG VM Agent**. To ensure that the "inside view" of Windows VMs is obtained using the eG VM Agent, set the **INSIDE VIEW USING** flag to **eG VM Agent (Windows)**. Once this is done, you can set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.
5. **DOMAIN, ADMIN USER, ADMIN PASSWORD,** and **CONFIRM PASSWORD** – By default, this test connects to each virtual guest remotely and attempts to collect "inside view" metrics. Accordingly, the **INSIDE VIEW USING** flag is set to **Remote connection to VM (Windows)** by default. To obtain a remote connection, the test must be configured with the privileges of an administrative user to the domain within which the guests reside. The first step towards this is to specify the **DOMAIN** within which the virtual guests reside. The **ADMIN USER** and **ADMIN PASSWORD** will change according to the **DOMAIN** specification. Discussed below are the different values that the **DOMAIN** parameter can take, and how they impact the **ADMIN USER** and **ADMIN PASSWORD** specifications:

   - **If the VMs belong to a single domain:** If the guests belong to a specific domain, then specify the name of that domain against the domain parameter. In this case, any administrative user in that

domain will have remote access to all the virtual guests. Therefore, an administrator account in the given domain can be provided in the **ADMIN USER** field and the corresponding password in the **ADMIN PASSWORD** field. Confirm the password by retyping it in the **CONFIRM PASSWORD** text box.

- **If the VMs belong to different domains:** In this case, you might want to provide multiple domain names. If this is done, then, to access the guests in every configured domain, the test should be configured with the required user privileges; this implies that along with multiple **DOMAIN** names, multiple **ADMIN USER** names and **ADMIN PASSWORDS** would also have to be provided. To help administrators provide these user details quickly and easily, the eG administrative interface embeds a special configuration page. To access this page, simply click on the **Click here** hyperlink that appears just above the parameters of this test in the test configuration page. To know how to use the special page, refer to Section 3.6.1 of this document.
- **If the INSIDE VIEW USING flag is set to 'eG VM Agent (Windows)':** On the other hand, if the **INSIDE VIEW USING** flag is set to **eG VM Agent (Windows)**, then it implies that the inside view can be obtained without domain administrator privileges. Therefore, set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.

6. **REPORT BY USER** – For the *Microsoft Hyper-V* monitoring model, the **REPORT BY USER** flag is set to **NO** by default, indicating that by default, the guest operating systems on the Hyper-V server are identified using the hostname specified in the operating system. On the other hand, for the *Microsoft Hyper-V VDI* model, this flag is set to **YES** by default; this implies that in case of VDI servers, by default, the guests will be identified using the login of the user who is accessing the guest OS. In other words, in VDI environments, this test will, by default, report measures for every *username_on_virtualmachinename*.
7. **REPORT POWERED OS** - **This flag becomes relevant only if the REPORT BY USER flag is set to 'Yes'.**

   If the **REPORT POWERED OS** flag is set to **Yes** (which is the default setting), then this test will report measures for even those VMs that do not have any users logged in currently. Such guests will be identified by their *virtualmachine name* and not by the *username_on_virtualmachinename*. On the other hand, if the **REPORT POWERED OS** flag is set to **No**, then this test will not report measures for those VMs to which no users are logged in currently.
8. **REPORT POWERED ON** - You can set the **REPORT POWERED ON** status to **Yes,** so that the test reports an additional measure, *Is VM powered on?*, revealing whether a guest OS is currently running or not. The default status of this flag is set to **Yes** for a *Hyper-V* server. For a *Hyper-V VDI* server on the other hand, the default status of this flag is **No**. This is because, in such environments, the virtual desktops will be in the powered-off state most of the time.
9. **IGNORE VMS INSIDE VIEW** - Administrators of some high security Hyper-V environments might not have permissions to internally monitor one/more VMs. The eG agent can be configured to **not obtain the 'inside view' of such 'inaccessible' VMs** using the **IGNORE VMS INSIDE VIEW** parameter. Against this parameter, you can provide a comma-separated list of VM names, or VM name patterns, for which the inside view need not be obtained. For instance, your **IGNORE VMS INSIDE VIEW** specification can be: **xp,*lin*,win*,vista*. Here, the * (asterisk) is used to denote leading and trailing

spaces (as the case may be). By default, this parameter is set to *none* indicating that the eG agent obtains the inside view of all VMs on a Hyper-V host by default.

Note:

While performing VM discovery, the eG agent will not discover the operating system of the VMs configured in the **IGNORE VMS INSIDE VIEW** text box.

10. **EXCLUDE VMS** - Administrators of some virtualized environments may not want to monitor some of their less-critical VMs - for instance, VM templates - both from 'outside' and from 'inside'. The eG agent in this case can be configured to completely exclude such VMs from its monitoring purview. To achieve this, provide a comma-separated list of VMs to be excluded from monitoring in the **EXCLUDE VMS** text box. Instead of VMs, VM name patterns can also be provided here in a comma-separated list. For example, your **EXCLUDE VMS** specification can be: **xp,*lin*,win*,vista*. Here, the * (asterisk) is used to denote leading and trailing spaces (as the case may be). By default, this parameter is set to *none* indicating that the eG agent obtains the inside and outside views of all VMs on a virtual host by default. By providing a comma-separated list of VMs/VM name patterns in the **EXCLUDE VMS** text box, you can make sure the eG agent stops collecting 'inside' and 'outside' view metrics for a configured set of VMs.
11. **IGNORE WINNT** – By default, the eG agent does not support the *inside view* for VMs executing on **Windows NT** operating systems. Accordingly, the **IGNORE WINNT** flag is set to **Yes** by default.
12. **DETAILED DIAGNOSIS** - To make diagnosis more efficient and accurate, the eG suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option.

    The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:

    - The eG manager license should allow the detailed diagnosis capability
    - Both the normal and abnormalfrequencies configured for the detailed diagnosis measures should not be 0.

Measures reported by the test:

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td>Is VM powered on? :</td><td>Whether the virtual machine is Hyper- V server host or not.</td><td></td><td>While the test reports a wide variety of other metrics too for virtual machines that are alive, only the Powered on status is indicated for virtual machines that are currently not available.

If this measure reports the value On, it indicates that the guest is up and running. The value Off could indicate that the guest has been powered- off; it could also indicate that the guest has moved to a different Hyper-V server.

The numeric values that correspond to each of the powered-on states discussed above are listed in the table below:
<table><tr><th>State</th><th>Value</th></tr><tr><td>On</td><td>1</td></tr><tr><td>Off</td><td>0</td></tr></table>
Note:

By default, this measure reports the values On or Off to indicate the status of a VM. The graph of this measure however, represents the status of a VM using the numeric equivalents - 0 or 1.</td></tr>
<tr><td>Virtual CPU allocated to VM</td><td>Indicates the number of processors currently present in this VM.</td><td>Number</td><td>All execution in the root and child partitions (where guest VMs run) happens on Virtual Processors (VPs). At a minimum, you will see one VP for each Logical Processor (LP). These account for the root VPs. You will then see one for each VP you have configured to a guest. Therefore, if you have an 8LP system with 1 guest running with 2 VPs, the count here will be 10.</td></tr>
<tr><td>Virtual CPU</td><td>Indicates the percentage</td><td>Percent</td><td>This measure serves as an effective</td></tr>
</table>

Measurement	Description	Measurement Unit	Interpretation
Utilization of VM	of time spent by the virtual processor assigned to this VM in guest and hypervisor code.		indicator of how resource- intensive a particular VM is on a specific Hyper-V server.
Virtual machine runtime	Indicates the percentage of time spent by the virtual processor in guest code.	Percent	Comparing the value of the *Virtual machine runtime* and *Hypervisor runtime* measures for every VM will reveal where the virtual processors of the VM have spent more time – in processing guest code or in processing hypervisor code?
Hypervisor runtime	Indicates the percentage of time the virtual processor spend in hypervisor code.	Percent	
Memory allocated to VM	Indicates the amount of physical memory currently allocated to this VM.	MB	
Data transmitted by VM	Indicates the number of bytes per second sent over the network adapters supported by this VM.	Mbps	
Data received by VM	Indicates the number of bytes per second sent over the network adapters supported by this VM.	Mbps	
Data dropped by VM	Indicates the number of bytes dropped on the network adapter since the last measurement period.	MB	Ideally, this value should be very low. A high value could be indicative of a network bottleneck.

Measurement	Description	Measurement Unit	Interpretation
Disk reads by VM	Indicates the number of bytes read per second from the disks attached to the IDE controller.	MB/Sec	These measures are good indicators of the activity on the disks attached to the IDE controller.
Disk writes by VM	Indicates the the number of bytes written per second to the disks attached to the IDE controller.	MB/Sec	
Deposited pages	Indicates the number of memory pages currently deposited into the partition.	Number	For each partition, the hypervisor maintains a memory pool of RAM SPA pages. This pool acts just like a checking account. The amount of pages in the pool is called the balancedeposited or withdrawn from the pool. When a hypercall that requires memory is made by a partition, the hypervisor withdraws the required memory from the total pool balance of that partition. When the balance available in the pool is less, then more memory pages are deposited in the pool. A very high value of this measure therefore, indicates that the balance in the pool maintained for this partition is dwindling. This is a cause for concern.
Hypercall	Indicates the rate of hypercalls made by this guest's code on the virtual processor.	Hypercalls/Sec	Hypercalls are one form of enlightenment. Guest OS's use the enlightenments to more efficiently use the system via the hypervisor. TLB flush is an example hypercall. If this value is zero, it is an indication that Integration Components are not installed. New OS's like WS08 can use hypercalls without enlightened drivers. So, hypercalls are only a prerequisite and not a guarantee for not having Integration Components installed.

Measurement	Description	Measurement Unit	Interpretation
Control register accesses	Indicates the rate of control register accesses by this guest on its virtual processors.	Accesses/Sec	Control registers are used to set up address mapping, privilege mode, etc.
HLT instructions	Indicates the rate of HLT instructions executed by t his guest on its virtual processors.	Instructions/Sec	A HLT will cause the hypervisor scheduler to de-schedule the current VP and move to the next VP in the runlist.
Emulated instruction	Indicates the rate of emulated instructions while executing guest code on the virtual processor.	Instructions/Sec	
MWAIT instructions	Indicates the rate of MWAIT instructions executed by this guest on its virtual processors.	Instructions/Sec	**The** MWAIT (monitored wait) instruction instructs the processor to enter a wait state in which the processor is instructed to monitor the address range between a and b and wait for an event or a store to that address range.
CPUID instructions	Indicates the rate of CPUID instructions executed by this guest on its virtual processors.	Instructions/Sec	The CPUID instruction is used to retrieve information on the local CPU's capabilities. Typically, CPUID is only called when the OS / Application first start. Therefore, this value is likely to be 0 most of the time.
Page fault intercepts	Indicates the rate of page fault exceptions intercepted by the hypervisor while executing this guest's code on the virtual processor	Intercepts/Sec	Whenever guest code accesses a page not in the CPU TLB a page fault will occur. This counter is closely correlated with the Large Page TLB Fills measure.
Total intercepts	Indicates the rate of	Intercepts/Sec	Whenever a guest VP needs to exit its

Measurement	Description	Measurement Unit	Interpretation
	hypervisor intercept messages.		current mode of running for servicing in the hypervisor, this is called an intercept. Some common causes of intercepts are resolving Guest Physical Address (GPA) to Server Physical Address (SPA) translations, privileged instructions like hlt / cupid / in / out, and the end of the VP's scheduled time slice.
Large page TLB fills	Indicates the rate of virtual TLB fills on large pages.	Fills/Sec	There are two types of TLB entries (and some three). Small TLB which generally means a 4K page and large Page which generally means 2MB. There are fewer Large TLB entries on the order of 8 – 32. A non- zero value for this measures indicates that the root partition is using large pages.
Small page TLB fills	Indicates the rate of virtual TLB fills on 4K pages.	Fills/Sec	There are two types of TLB entries (and some three). Small TLB which generally means a 4K page and large Page which generally means 2MB. There are fewer Large TLB entries on the order of 64 – 1024+.
Cpu utilization of VM	Indicates the percentage of allocated CPU resources that this VM is currently using.	Percent	Comparing the value of this measure across VMs will enable you to accurately identify the VMs on which CPU-intensive applications are executing.
Disk capacity of VM	Indicates the total disk capacity of the VM.	MB	Since VMs are easy to create and deploy, many a time an administrator might be faced with scenarios where many VMs are created on an Hyper-V host, but very few are actively used. A VM, whether powered on or off, consumes disk space on a host. When the Hyper-V server hosting the VMs runs low on disk space, administrators might want to know which VM is taking up

Measurement	Description	Measurement Unit	Interpretation
			maximum disk space. This measure reveals the disk capacity of a VM, regardless of its on/off state. A quick comparison of the capacity across VMs can enable administrators to accurately identify the VM that is taking up maximum disk space.
Disk reads and writes by VM	Indicates the rate at which read- write requests were processed by this VM.	Kbytes/Sec	Compare the value of this measure across VMs to know on which VM I/O activity was abnormally high.
Data sent and received by VM	Indicates the rate at which network I/O is processed by this VM.	Mbps	Compare the value of this measure across VMs to know on which VM network I/O activity was abnormally high.
VM health	Indicates the current state of this VM.	Number	If the value reported by this measure is 1, the status is Ok. If the value reported by this measure is 3, the status is Critical. The Detailed Diagnosis (DD) of this measure shows the VM State, Process ID, and Operational Status.

3.6.1.1 Configuring Users for VM Monitoring

In order to enable the eG agent to connect to VMs in multiple domains and pull out metrics from them, the eG administrative interface provides a special page using which the different **DOMAIN** names, and their corresponding **ADMIN USER** names and **ADMIN PASSWORDS** can be specified. To access this page, just click on the **Click here** hyperlink in any of the VM test configuration pages.

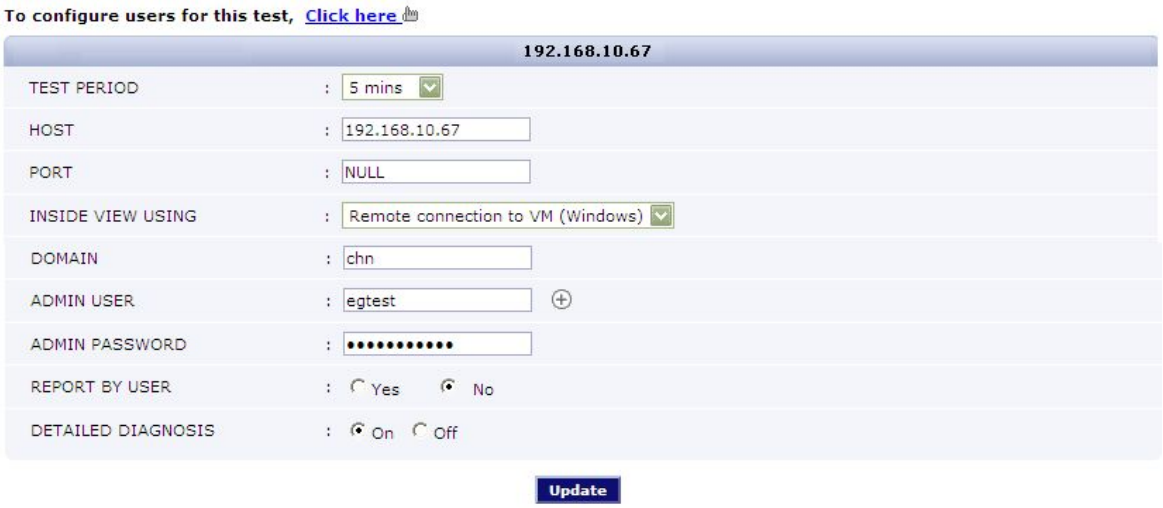


Figure 3.9: Figure 2.9: Configuring a VM test

Upon clicking, Figure 3.10 will appear, using which the VM user details can be configured.

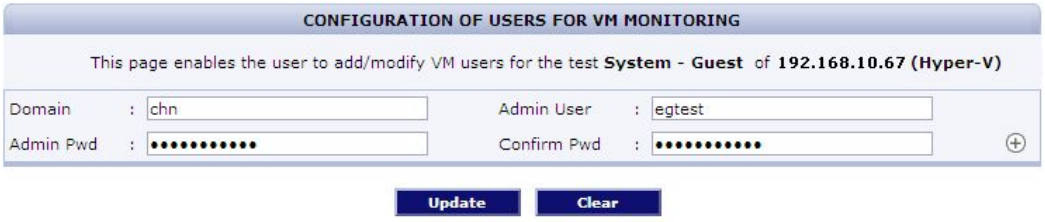


Figure 3.10: Figure 2.10: The VM user configuration page

To add a user specification, do the following:

a. First, provide the name of the **Domain** to which the VMs belong (see Figure 3.10). If one/more VMs do not belong to any domain, then, specify *none* here.

The eG agent must be configured with user privileges that will allow the agent to communicate with the VMs in a particular domain and extract statistics. If *none* is specified against **Domain**, then a local user account can be provided against **Admin User**. On the other hand, if a valid **Domain** name has been specified, then a domain administrator account can be provided in the **Admin User** text box.

The password of the specified **Admin User** should be mentioned in the **Admin Pwd** text box.

Confirm the password by retyping it in the **Confirm Pwd** text box.

To add more users, click on the ⊕ button in Figure 3.10. This will allow you to add one more user specification as depicted by Figure 3.11.

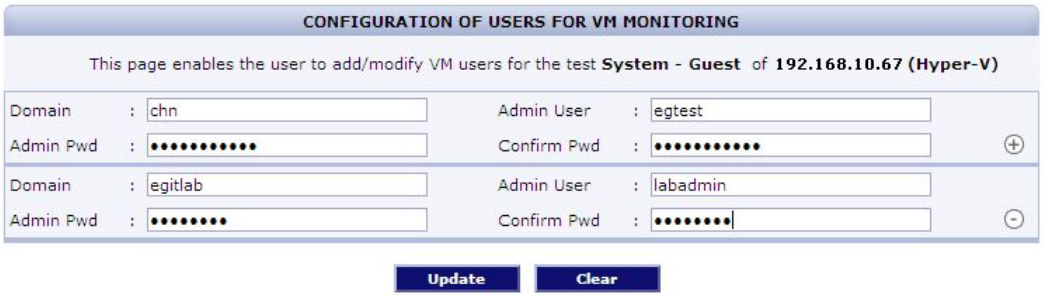


Figure 3.11: Figure 2.11: Adding another user

In some virtualized environments, the same **Domain** could be accessed using multiple **Admin User** names. For instance, to login to a **Domain** named *egitlab*, the eG agent can use the **Admin User** name *labadmin* or the **Admin User** name *jadmn*. You can configure the eG agent with the credentials of both these users as shown by Figure 3.12.

The same 'Domain' mapped to different 'Admin Users'

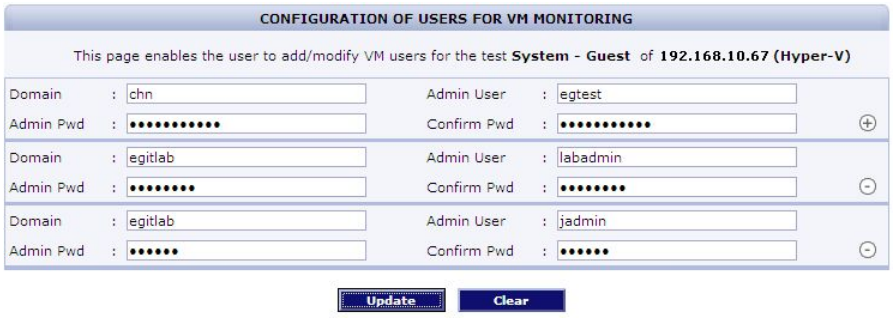


Figure 3.12: Figure 2.12: Associating a single domain with different admin users

When this is done, then, while attempting to connect to the domain, the eG agent will begin by using the first **Admin User** name of the specification. In the case of Figure 3.12, this will be *labadmin*. If, for some reason, the agent is unable to login using the first **Admin User** name, then it will try to login again, but this time using the second **Admin User** name of the specification – i.e., *jadmn* in our example (see Figure 3.12). If the first login attempt itself is successful, then the agent will ignore the second **Admin User** name.

To clear all the user specifications, simply click the **Clear** button in Figure 3.12.

To remove the details of a particular user alone, just click the ⊖ button in Figure 3.12.

To save the specification, just click on the **Update** button in Figure 3.12. This will lead you back to the test configuration page, where you will find the multiple domain names, user names, and passwords listed against the respective fields (see Figure 3.13).

Figure 3.13: Figure 2.13: The test configuration page displaying multiple domain names, user names, and passwords

3.6.2 Hyper-V VM Information Test

Hyper-V™ live migration is designed to move running VMs with no impact on VM availability to users. By pre-copying the memory of the migrating VM to the destination physical host, live migration minimizes the amount of transfer time of the VM A live migration is deterministic, meaning that the administrator, or script, that initiates the live migration can control which computer is the destination for the live migration. The guest operating system in the migrating VM is unaware that the migration is happening, so no special configuration for the guest operating system is needed.

Below is a summary of the live migration process:

- All VM memory pages are transferred from the source Hyper-V™ physical host to the destination Hyper-V™ physical host. While this is occurring, any VM modifications to its memory pages are tracked.
- ™Pages that were modified while step 1 was occurring are transferred to the destination physical computer.
- The storage handle for the VM's VHD files are moved to the destination physical computer.
- The destination VM is brought online on the destination Hyper-V™ server.

This test reports the number of guests registered with the server, and promptly alerts administrators to addition/removal of guests from the server.

Target of the test:

A Hyper-V server

Agent executing the test:

An internal agent

Output of the test

One set of results for the Hyper-V server monitored

Parameters of the test:

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured.
3. **PORT** - The port at which the **HOST** listens. By default, this is NULL.
4. **INSIDE VIEW USING** - By default, this test communicates with every VM remotely and extracts "inside view" metrics. Therefore, by default, the **INSIDE VIEW USING** flag is set to **Remote connection to VM (Windows)**.

   Typically, to establish this remote connection, eG Enterprise requires that the eG agent be configured with domain administrator privileges. In high-security environments, where the IT staff might have reservations about exposing the credentials of their domain administrators, this approach to extracting "inside view" metrics might not be preferred. In such environments therefore, eG Enterprise provides administrators the option to deploy a piece of software called the **eG VM Agent** on every Windows VM; this VM agent allows the eG agent to collect "inside view" metrics from the Windows VMs **without domain administrator rights.** Refer to Section 1.3.2 for more details on the **eG VM Agent**. To ensure that the "inside view" of Windows VMs is obtained using the eG VM Agent, set the **INSIDE VIEW USING** flag to **eG VM Agent (Windows)**. Once this is done, you can set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.
5. **DOMAIN, ADMIN USER, ADMIN PASSWORD,** and **CONFIRM PASSWORD** – By default, this test connects to each virtual guest remotely and attempts to collect "inside view" metrics. Accordingly, the **INSIDE VIEW USING** flag is set to **Remote connection to VM (Windows)** by default. To obtain a remote connection, the test must be configured with the privileges of an administrative user to the domain within which the guests reside. The first step towards this is to specify the **DOMAIN** within which the virtual guests reside. The **ADMIN USER** and **ADMIN PASSWORD** will change according to the **DOMAIN** specification. Discussed below are the different values that the **DOMAIN** parameter can take, and how they impact the **ADMIN USER** and **ADMIN PASSWORD** specifications:

   - **If the VMs belong to a single domain:** If the guests belong to a specific domain, then specify the name of that domain against the domain parameter. In this case, any administrative user in that domain will have remote access to all the virtual guests. Therefore, an administrator account in the given domain can be provided in the **ADMIN USER** field and the corresponding password in the **ADMIN PASSWORD** field. Confirm the password by retyping it in the **CONFIRM PASSWORD** text box.

- **If the VMs belong to different domains:**In this case, you might want to provide multiple domain names. If this is done, then, to access the guests in every configured domain, the test should be configured with the required user privileges; this implies that along with multiple **DOMAIN** names, multiple **ADMIN USER** names and **ADMIN PASSWORDS** would also have to be provided. To help administrators provide these user details quickly and easily, the eG administrative interface embeds a special configuration page. To access this page, simply click on the **Click here** hyperlink that appears just above the parameters of this test in the test configuration page. To know how to use the special page, refer to Section 3.6.2 of this document.
- **If the INSIDE VIEW USING flag is set to 'eG VM Agent (Windows)':** On the other hand, if the **INSIDE VIEW USING** flag is set to **eG VM Agent (Windows)**, then it implies that the inside view can be obtained without domain administrator privileges. Therefore, set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.

6. **REPORT BY USER** – For the *Microsoft Hyper-V* monitoring model, the **REPORT BY USER** flag is set to **NO** by default, indicating that by default, the guest operating systems on the Hyper-V server are identified using the hostname specified in the operating system. On the other hand, for the *Microsoft Hyper-V VDI* model, this flag is set to **YES** by default; this implies that in case of VDI servers, by default, the guests will be identified using the login of the user who is accessing the guest OS. In other words, in VDI environments, this test will, by default, report measures for every *username_on_ virtualmachinename*.
7. **REPORT POWERED OS** - **This flag becomes relevant only if the REPORT BY USER flag is set to 'Yes'.**

   If the **REPORT POWERED OS** flag is set to **Yes** (which is the default setting), then this test will report measures for even those VMs that do not have any users logged in currently. Such guests will be identified by their *virtualmachine name* and not by the *username_on_virtualmachinename*. On the other hand, if the **REPORT POWERED OS** flag is set to **No**, then this test will not report measures for those VMs to which no users are logged in currently.
8. **REPORT POWERED ON** - You can set the **REPORT POWERED ON** status to **Yes,** so that the test reports an additional measure, *Is VM powered on?*, revealing whether a guest OS is currently running or not. The default status of this flag is set to **Yes** for a *Hyper-V* server. For a *Hyper-V VDI* server on the other hand, the default status of this flag is **No**. This is because, in such environments, the virtual desktops will be in the powered-off state most of the time.
9. **IGNORE VMS INSIDE VIEW** - Administrators of some high security Hyper-V environments might not have permissions to internally monitor one/more VMs. The eG agent can be configured to **not obtain the 'inside view' of such 'inaccessible' VMs** using the **IGNORE VMS INSIDE VIEW** parameter. Against this parameter, you can provide a comma-separated list of VM names, or VM name patterns, for which the inside view need not be obtained. For instance, your **IGNORE VMS INSIDE VIEW** specification can be: **xp,*lin*,win*,vista*. Here, the * (asterisk) is used to denote leading and trailing spaces (as the case may be). By default, this parameter is set to *none* indicating that the eG agent obtains the inside view of all VMs on a Hyper-V host by default.

Note:

While performing VM discovery, the eG agent will not discover the operating system of the VMs configured in the **IGNORE VMS INSIDE VIEW** text box.

10. **EXCLUDE VMS** - Administrators of some virtualized environments may not want to monitor some of their less-critical VMs - for instance, VM templates - both from 'outside' and from 'inside'. The eG agent in this case can be configured to completely exclude such VMs from its monitoring purview. To achieve this, provide a comma-separated list of VMs to be excluded from monitoring in the **EXCLUDE VMS** text box. Instead of VMs, VM name patterns can also be provided here in a comma-separated list. For example, your **EXCLUDE VMS** specification can be: **xp,*lin*,win*,vista*. Here, the * (asterisk) is used to denote leading and trailing spaces (as the case may be). By default, this parameter is set to *none* indicating that the eG agent obtains the inside and outside views of all VMs on a virtual host by default. By providing a comma-separated list of VMs/VM name patterns in the **EXCLUDE VMS** text box, you can make sure the eG agent stops collecting 'inside' and 'outside' view metrics for a configured set of VMs.
11. **IGNORE WINNT** – By default, the eG agent does not support the *inside view* for VMs executing on **Windows NT** operating systems. Accordingly, the **IGNORE WINNT** flag is set to **Yes** by default.
12. **DD FREQUENCY** - Refers to the frequency with which detailed diagnosis measures are to be generated for this test. For a *Microsoft Hyper-V* server, this is set to *1:1* by default. This indicates that, by default, detailed measures will be generated every time this test runs, and also every time the test detects a problem. It is recommended that you do not change the default setting of this parameter. This is because, eG Enterprise can discover the IP addresses of the guest operating systems on a Hyper-V host, only while generating the detailed measures for this test. The automatic discovery of the guest IPs, in turn, enables eG Enterprise to perform **AutoVirtualMapping**.
13. **DETAILED DIAGNOSIS** - To make diagnosis more efficient and accurate, the eG suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option.

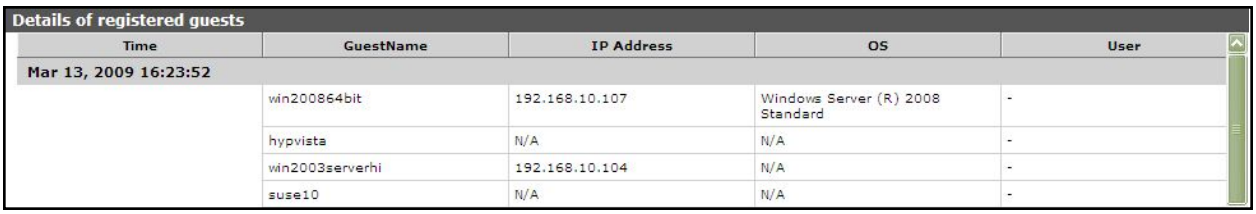
    The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:

    - The eG manager license should allow the detailed diagnosis capability
    - Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
Registered VMs	Indicates the total number of virtual machines that have been registered with the server currently.	Number	
VMs powered on	Indicates the number of guests that are currently powered on.	Number	To know which are the guests that are powered on, use the detailed diagnosis capability of this measure (if enabled).
VMs with users	Indicates the number of powered on guests with users logged in currently.	Number	To know which guests the users have logged into, use the detailed diagnosis capability of this measure (if enabled). **Note that this measure will not be available for the ' Microsoft Hyper-V' server model.**
VMs without users	Indicates the number of powered on guests without any users logged in currently.	Number	**Note that this measure will not be available for the 'Microsoft Hyper-V' server model.**
Added VMs	Indicates the number of guests that were newly added to the server during this measurement period.	Number	The detailed diagnosis of these measures, if enabled, lists the virtual machines that were migrated to or from (as the case may be) the Hyper-V server.
Removed VMs	Indicates the number of guests that were newly removed from the server during this measurement period.	Number	

The detailed diagnosis of the *Registered VMs* measure reports the name of the guests registered with the Hype-V server, the IP address of the guests, the guest OS, and the name of the user currently logged into the guest.



Details of registered guests

Time	GuestName	IP Address	OS	User
Mar 13, 2009 16:23:52				
	win200864bit	192.168.10.107	Windows Server (R) 2008 Standard	-
	hypvista	N/A	N/A	-
	win2003serverhi	192.168.10.104	N/A	-
	suse10	N/A	N/A	-

Figure 3.14: The detailed diagnosis of the Registered guests measure

The detailed diagnosis of the *VMs powered on* measure reports the name of the guests currently powered on, the IP address of the guests, the guest OS, and the name of the user currently logged into the guest.

Details of guests powered on

Time	GuestName	IP Address	OS	User
Mar 13, 2009 16:23:52				
	win200864bit	192.168.10.107	Windows Server (R) 2008 Standard	-
	hypvista	N/A	N/A	-
	win2003serverhi	192.168.10.104	N/A	-

Figure 3.15: The detailed diagnosis of the Guests powered on measure

Note:

The eG agent can extract the name and "outside view" metrics of Linux guests, but can neither discover the IP address nor report "inside view" metrics pertaining to Linux guests. Similarly, the eG agent cannot discover the IP address or obtain the "inside view" of those Windows VMs which do not support **Key/Value Pair Exchange** script

3.6.3 Virtual Machine Management Service Test

The Virtual Machine Management Service (VMMS) is responsible for managing the state of all virtual machines in child partitions. By periodically monitoring the VMMS, you can exercise better control over the operations of the VMs. This test monitors the VMMS and reports the number of VMs in various states.

Target of the test:

A Hyper-V / Hyper-V VDI server

Agent executing the test:

An internal agent

Output of the test:

One set of results for the Hyper-V server monitored

Parameters of the test:

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
Deleting	Indicates the number of virtual machines that are currently being deleted.	Number	
Exporting	Indicates the number of VMs that are currently exporting	Number	
Merging disks	Indicates the number of virtual machines that are merging disks currently.	Number	
Paused	Indicates the number of virtual machines that have been paused currently.	Number	
Running	Indicates the number of virtual machines that are currently running.	Number	
Turned off	Indicates the number of virtual machines that are currently turned off.	Number	

3.6.4 VM Connectivity Test

Sometimes, a VM could be in a powered-on state, but the failure of the VM operating system or any fatal error in VM operations could have rendered the VM inaccessible to users. In order to enable administrators to promptly detect such 'hidden' anomalies, the eG agent periodically runs a connectivity check on each VM using the VM Connectivity test, and reports whether the VM is accessible over the network or not.

Target of the test:

A Hyper-V / Hyper-V VDI server

Agent executing the test:

An internal agent

Output of the test:

One set of results for each VM configured on the Hyper-V host being monitored

Parameters of the test:

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured.
3. **PORT** - The port at which the specified **HOST** listens. By default, this is NULL.
4. **PACKETSIZE** - The size of packets used for the test (in bytes)
5. **PACKETCOUNT** – The number of packets to be transmitted during the test
6. **TIMEOUT** - How long after transmission should a packet be deemed lost (in seconds)
7. **PACKETINTERVAL** - Represents the interval (in milliseconds) between successive packet transmissions during the execution of the network test for a specific target.
8. **REPORTUNAVAILABILITY** – By default, this flag is set to **No**. This implies that, by default, the test will not report the unavailability of network connection to any VM. In other words, if the *Network availability of VM* measure of this test registers the value 0 for any VM, then, by default, this test will not report any measure for that VM; under such circumstances, the corresponding VM name will not appear as a descriptor of this test. You can set this flag to **Yes**, if you want the test to report and alert you to the unavailability of the network connection to a VM.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
Avg network delay	Indicates the average delay between transmission of packet to a VM and receipt of the response to the packet at	Secs	An increase in network latency could result from misconfiguration of the router (s) along the path, network congestion, retransmissions at the

Measurement	Description	Measurement Unit	Interpretation
	the source.		network, etc.
Min network delay	The minimum time between transmission of a packet and receipt of the response back.	Secs	A significant increase in the minimum round-trip time is often a sure sign of network congestion.
Packet loss	Indicates the percentage of packets lost during transmission from source to target and back.	Percent	Packet loss is often caused by network buffer overflows at a network router or by packet corruptions over the network. The detailed diagnosis for this measure provides a listing of routers that are on the path from the external agent to target server, and the delays on each hop. This information can be used to diagnose the hop(s) that could be causing excessive packet loss/delays.
Network availability of VM	Indicates whether the network connection is available or not.	Percent	A value of 100 indicates that the VM is connected. The value 0 indicates that the VM is not connected. Typically, the value 100 corresponds to a Packet loss of 0.

3.6.5 Hyper-V Dynamic Memory for VMs

Dynamic Memory is a new Hyper-V feature that helps you to use physical memory more efficiently. With Dynamic Memory, Hyper-V treats memory as a shared resource that can be reallocated automatically among running virtual machines. Dynamic Memory adjusts the amount of memory available to a virtual machine, based on changes in memory demand and values that you specify.

Using this test, you can determine whether or not the Dynamic Memory feature is enabled on a virtual machine, and if so, assess how well that feature works. In the process, you can also ascertain the following:

- Isolate resource-hungry VMs;
- Understand the Dynamic Memory configuration of each VM;
- Figure out whether this configuration needs to be fine-tuned to facilitate more efficient and effective resource-sharing among VMs.

Target of the test:

A Hyper-V / Hyper-V VDI server

Agent executing the test:

An internal agent

Output of the test:

One set of results for the each VM on the Hyper-V host monitored

Parameters of the test:

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
Is dynamic memory enabled?	Indicates whether the Dynamic Memory feature is enabled or not in this VM.		There is no global setting to turn Dynamic Memory on or off at the host level. It must be configured for each virtual machine. By default, a virtual machine is set up with the traditional static amount of memory. You can edit the properties of a virtual machine to enable Dynamic Memory. **`This measure reports the value`** *Yes* if Dynamic Memory is enabled on a

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td></td><td></td><td></td><td>VM, and the value No if it is not.

The tabe below lists the numeric values that correspond to the Yes/No values reported by the measure:

<table><tr><th>State</th><th>Value</th></tr><tr><td>Yes</td><td>1</td></tr><tr><td>No</td><td>0</td></tr></table>
Note:

By default, this measure reports the values Yes or No to indicate whether dynamic memory is enabled or not for a measure. The graph of this measure however, represents the dynamic memory status using the numeric equivaluents - 0 or 1 - only.</td></tr>
<tr><td>Added memory</td><td>Indicates the amount of physical memory added to this VM.</td><td>MB</td><td></td></tr>
<tr><td>Removed memory</td><td>Indicates the amount of physical memory removed from this VM.</td><td>MB</td><td></td></tr>
<tr><td>Physical memory</td><td>Indicates the amount of physical memory allocated to this VM.</td><td>MB</td><td></td></tr>
<tr><td>Guest visible physical memory</td><td>Indicates the amount of physical memory actually utilized by this VM as seen from within the VM.</td><td>MB</td><td></td></tr>
<tr><td>Average pressure</td><td>Indicates the average memory pressure on this VM.</td><td>Percent</td><td>Dynamic Memory determines the amount of memory needed by a virtual machine by calculating something called memory pressure. To perform this calculation, Hyper-V looks at the total committed memory of the guest operating system running in the virtual machine and then</td></tr>
</table>

Measurement	Description	Measurement Unit	Interpretation
			calculates pressure as the ratio of how much memory the virtual machine wants to how much it has. A very high value of this measure therefore indicates that the VM is resource-hungry.
Current pressure	Indicates the current memory pressure in this VM.	Percent	
Maximum pressure	Indicates the maximum pressure band in this VM.	Percent	
Minimum pressure	Indicates the minimum pressure band in this VM.	Percent	
Memory add operations	Indicates the number of memory addition operations performed on this VM since the last measurement period.	Number	A consistent rise in the value of this measure could indicate that the memory needs of the VM are growing.
Memory remove operations	Indicates the number of memory removal operations performed on this VM since the last measurement period.	Number	If Dynamic Memory sees pressure reduce within a VM, it is an indication that memory can be returned back to the pool, making it available for reassignment. To remove unneeded memory from a Dynamic Memory- enabled VM, Hyper- V uses a process called ballooning. Using a balloon, Dynamic Memory effectively blocks the memory freed up by a VM. This means that the VM cannot use the memory until the balloon (the block) is shrunk by being re-assigned that memory from the available memory on the host. Once the balloon is in place, the Dynamic Memory works with the parent partition to reassign the physical memory back to the host.
Starting memory	Indicates the total amount of RAM in the virtual system, as	MB	The value of this measure needs to be high enough to allow the guest operating

Measurement	Description	Measurement Unit	Interpretation
	seen by this guest operating system. For a virtual system with dynamic memory enabled, this represents the initial memory available at startup.		system to start, but should be as low as possible to allow for optimal memory utilization and potentially higher consolidation ratios.
Maximum memory	Indicates the maximum amount of memory that may be consumed by this VM. For a virtual system with dynamic memory enabled, this represents the maximum memory setting.	MB	The value can be set from as low as the value for Startup RAM to as high as 64 GB. However, a virtual machine can use only as much memory as the maximum amount supported by the guest operating system. For example, if you specify 64 GB for a virtual machine running a guest operating system that supports a maximum of 32 GB, the virtual machine cannot use more than 32 GB.
Memory buffer	Defines the amount of extra memory that should be reserved for this virtual machine at runtime, as a percentage of the total memory that the virtual machine is thought to need.	Percent	Memory buffer is specified as a percentage because the actual amount of memory that represents the buffer changes in response to changes in memory usage while the virtual machine is running. Hyper-V uses performance counters in the virtual machine that identify committed memory to determine the current memory requirements of the virtual machine and then calculates the amount of memory to add as a buffer. The buffer is determined using the following formula: Amount of memory buffer = how much memory the virtual machine actually needs / (memory buffer value / 100). For example, if the memory committed to the guest operating system is 1000 MB and the memory buffer is 20%, Hyper-V will attempt to allocate an additional 20% (200 MB) for a total of 1200 MB of physical memory allocated to the virtual machine.

Measurement	Description	Measurement Unit	Interpretation
Memory weight	Defines the memory allocation weighting value for this virtual machine. After all reserves have been met, the remaining memory of the hosting platform will be allocated to virtual systems based on their relative weights (not to exceed the value specified by the Limit property). This property is inherited from CIM_ ResourceAllocationSettingData.	Number	This provides Hyper-V with a way to determine how to distribute memory among VMs if there is not enough physical memory available on the host to give every VM the amount of memory it requests.
Memory demand	Indicates the amount of memory that this VM needs to perform correctly, as per the running workload.	MB	A high value indicates that the VM is processing a memory-intensive workload.

3.6.6 Hyper-V VM Heartbeat Status Test

User access to a VM can be disrupted by many factors. A poor network link or a broken network link can delay/deny users access to a VM. Beside such external network connectivity issues, a user may not be able to reach a VM owing to internal issues as well – these issues can range from a VM lock, a VM crash, or a sudden termination of a VM's operations. This is why, when a user complains of being unable to access a VM, the administrator needs to quickly determine the reason for the inaccessibility of the VM, so that the correct remedial action can be initiated and access to the VM can be swiftly restored.

The **Hyper-V VM Heartbeat Status** test periodically monitors the hearbeat service installed on each VM and reports whether that service and the VM it is operating on are functioning properly or not. The heartbeat service allows the parent partition to detect when a virtual machine has locked up, crashed or otherwise ceased to function. The parent partition sends heartbeat messages to the guest operating system at regular intervals. It is then the job of the Hyper-V Heartbeat Service installed on the guest operating system to send a response to each of these heartbeat messages. When the parent partition fails to receive responses from the child partition, it assumes that the child's Heartbeat Service, and therefore the guest operating system on which it is running, has encountered problems. By closely monitoring the heartbeat service, this test enables administrators to determine whether/not internal issues (eg., a VM lock, a VM crash, etc.) are affecting the accessibility of a VM. If the test reports that the heartbeat service and the VM it is installed on are up and running, the administrator can safely conclude that internal factors are not responsible for the unavailability of that VM; further investigation as to the reason for the VM's unavailability can then be carried out.

Target of the test:

A Hyper-V / Hyper-V VDI server

Agent executing the test:

An internal agent

Output of the test:

One set of results for the Hyper-V host monitored

Parameters of the test:

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured.
3. **DETAILED DIAGNOSIS** - To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option.

   The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:

   - The eG manager license should allow the detailed diagnosis capability
   - Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
OK status	Indicates the number of VMs on which the heartbeat service is operating normally.	Number	A high value is desired for this measure. A low value indicates that the parent partition is unable to communicate with the heartbeat service on many VMs; this in turn implies that many VMs are currently

Measurement	Description	Measurement Unit	Interpretation
			unreachable. If this is the case, you will have to figure out why those VMs are unavailable and initiate the required corrective action. You can use the detailed diagnosis of this measure to know which VMs are operating normally.
Error status	Indicates the number of VMs that do not support a compatible protocol version.	Number	Ideally, the value of this measure should be low. You can use the detailed diagnosis of this measure to know which VMs have encounrtered errors.
Lost contact status	Indicates the number of VMs on which the heartbeat service has not been installed yet or has not yet been contacted by the parent partition.	Number	Ideally, the value of this measure should be 0. You can use the detailed diagnosis of this measure to know on which VMs the heartbeat service has not been installed or is yet to be contacted.
Lost communication status	Indicates the number of VMs on which the hearbeat service is not responding to the hearbeat messages sent by the parent partition.	Number	Ideally, the value of this measure should be 0. A high value indicates that the heartbeat service on many VMs is not responding to heartbeat messages. This could be owing to a VM lock, a VM crash, or any other activity that can temporarily/permanently suspend VM operations. You can use the detailed diagnosis of this measure to know the VMs with which the parent partition is unable to communicate.
Unknown status	Indicates the number of VMs that have been powered off.	Number	If a VM is powered off, the parent partition will not be able to contact the heartbeat service on that VM at

Measurement	Description	Measurement Unit	Interpretation
			all. This again can cause user accesses to that VM to be denied. You can use the detailed diagnosis of this measure to know the VMs that are in an Unknown state.

The detailed diagnosis of the *OK status* measure reveals the VMs that are currently operating normally.

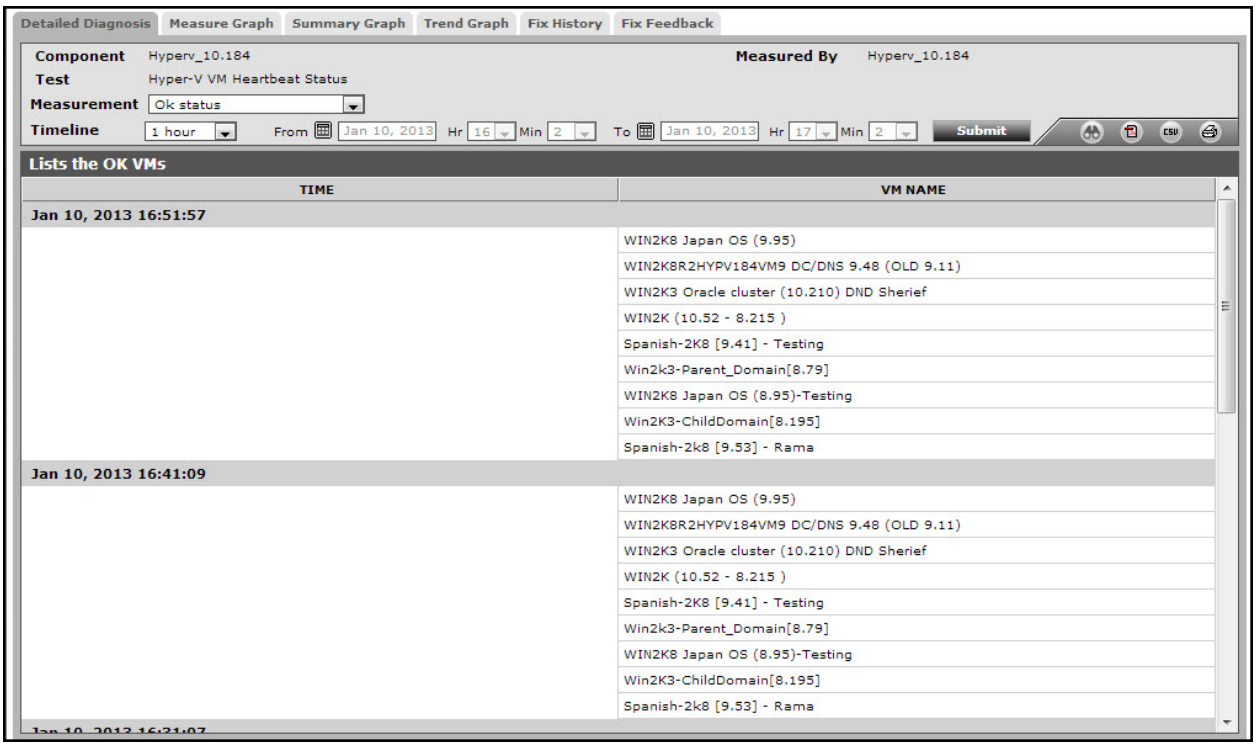


Figure 3.16: The detailed diagnosis of the OK status measure

3.6.7 Hyper-V VM Replications Test

Hyper-V Replica enables organizations to implement an affordable Business Continuity and Disaster Recovery (BCDR) solution for virtualized workloads. This allows virtual machines running at a primary site to be efficiently replicated to secondary location (Replica site) across a WAN link.

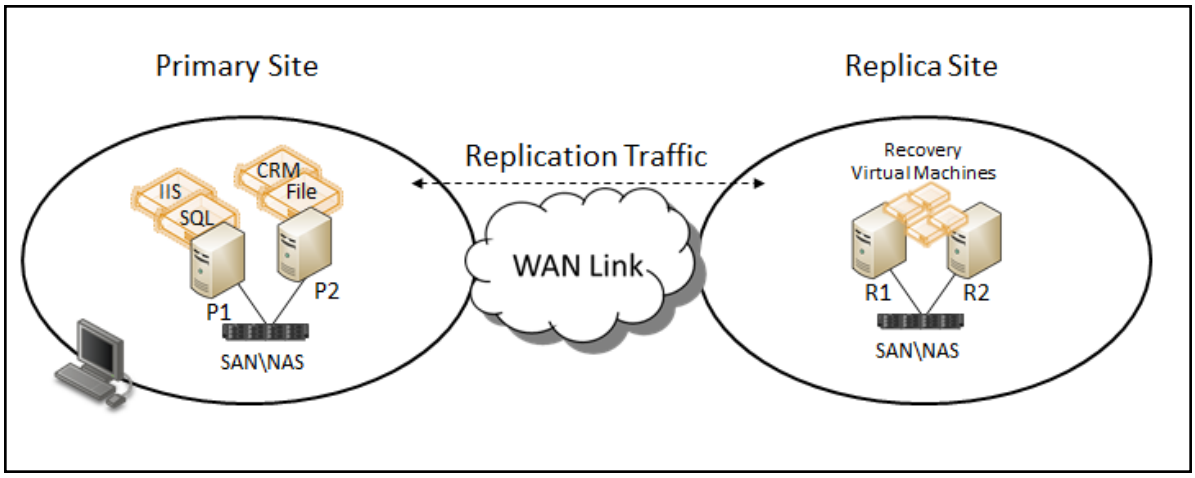


Figure 3.17: How Hyper-V Replica works

Delays/errors in the replication process can cause a severe data non-sync between the primary and secondary sites, resulting in significant loss of data when disaster strikes and recovery is attempted. To protect the data from loss, you need to monitor the replication machinery continuously, and on the slightest sign of a disturbance, alert the relevant administrators and ensure that the anomaly is promptly remediated. The **Hyper-V VM Replication** test does just that. This test monitors the replication activity performed by Hyper-V Replica for each VM on a Hyper-V host, instantly detects latencies or inconsistencies in the process, and proactively warns administrators of the same, so that the necessary corrective/control action can be taken.

Note:

This test will report metrics only for Microsoft Hyper-V 2012.

Note:

This test will report metrics for Microsoft Hyper-V Server 2012 only.

Target of the test:

A Hyper-V / Hyper-V VDI server

Agent executing the test:

An internal agent

Output of the test:

One set of results for each VM on a Hyper-V host monitored

Parameters of the test:

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
Average replication latency	Indicates the average time taken to replicate this VM to another host.	Secs	A low value is desired for this measure. Typically, replication is said to be 'Normal' if latency is less than 5 minutes. A high value indicates that too much time is taken for replicating a VM. This could be owing to network connectivity issues, storage issues on the primary or replica or if the primary VM requires resynchronization. This is a cause for concern, as it can cause significant data loss at the time of a fail-over. To identify the VM that is taking the maximum time to replicate, compare the value of this measure across VMs.
Average replication size	Indicates the average size of the replication files related to this VM.	MB	If large sized files are transferred over the network as part of a replication activity, it is bound to consume more bandwidth and even delay the replication process. A low value is hence desired for this measure per VM. To find out which VM's replica contains files of the maximum size, compare the value

Measurement	Description	Measurement Unit	Interpretation
			of this measure across VMs.
Compression efficiency	Indicates the percentage compression efficiency for the files that have been transferred over the network when replicating this VM.	Percent	Higher the value of this measure, greater the compression efficiency. This in turn implies optimal bandwidth usage over the network. A low value hence indicates that too much bandwidth is used when transferring replicated files over the network. Its good practice to c onfigure Hyper- V Replica to compress the data transmitted over the network in the settings for the virtual machine in Hyper-V Manager. You can also use tools outside of Hyper-V to perform compression.
Last replication size	Indicates the size of the files replicated for this VM during the last replication.	MB	
Network data received	Indicates the data received for this VM over the network since the virtual machine management service was started.	MB	
Network data sent	Indicates the data sent for this VM over the network since the virtual machine management service was started.	MB	
Replication count	Indicates the number of replication cycles that have run for this VM since the virtual machine management service was started.	Number	
Replication latency	Indicates the last replication latency of this VM.	Secs	It is the time taken for the delta to be

Measurement	Description	Measurement Unit	Interpretation
			applied on the recovery since it was snapped. A low value is desired for this measure.
Resynchronized data	Indicates the data sent and received over the network for this VM during the resynchronize operation since the virtual machine management service was started.	MB	A resynchronization essentially compares blocks between the Primary and Replica VHDs and then sends the delta blocks to the Replica. Scenarios where this can happen include, but may not be limited to, a failure occurred on the Primary server when changes were being made to the replication log or, if the Primary is a Failover Cluster, an unplanned cluster failover occurred.

3.6.8 Hyper-V VM Replication Health Status Test

Windows Server 2012 Hyper-V Role introduces a new capability, Hyper-V Replica, as a built-in replication mechanism at a virtual machine (VM) level. Hyper-V Replica can asynchronously replicate a selected VM running at a primary site to a designated replica site across LAN/WAN.

To track the replication status of each VM and promptly capture errors in the replication process, administrators can use the Hyper-V VM Replication Health Status test.

Note:
This test will report metrics for Microsoft Hyper-V Server 2012 only.

Target of the test:

A Hyper-V / Hyper-V VDI server

Agent executing the test:

An internal agent

Output of the test:

One set of results for each VM on a Hyper-V host monitored

Parameters of the test:

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured.
3. **DETAILED DIAGNOSIS** - To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option.

   The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:

   - The eG manager license should allow the detailed diagnosis capability
   - Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measures reported by the test:

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td>Replication state</td><td>Indicates the replication state of this VM.</td><td></td><td>The values that this measure can report and their corresponding numeric values are as follows:
<table>
<tr><th>Measure Value</th><th>Numeric Value</th></tr>
<tr><td>Error</td><td>0</td></tr>
<tr><td>FailOverWaitingCompletion</td><td>1</td></tr>
<tr><td>FailedOver</td><td>2</td></tr>
<tr><td>NotApplicable</td><td>3</td></tr>
<tr><td>ReadyForInitialReplication</td><td>4</td></tr>
<tr><td>Replicating</td><td>5</td></tr>
<tr><td>Resynchronizing</td><td>6</td></tr>
<tr><td>ResynchronizeSuspended</td><td>7</td></tr>
<tr><td>Suspended</td><td>8</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><td>SyncedReplicationComplete</td><td>9</td></tr>
<tr><td>WaitingForInitialReplication</td><td>10</td></tr>
<tr><td>WaitingForStartResynchronize</td><td>11</td></tr>
</table>
Note:
By default, this measure reports the Measure Values in the table above to indicate replication state. In the graph of this measure however, the same will be re presented using the numeric equivalents only.
The detailed diagnosis of this measure reports the primary server name, the replica server name, and the current replica server name of a VM.
</td></tr>
<tr><td>Replication health</td><td>Indicates the replication health of this VM.</td><td></td><td>The values that this measure can report and their corresponding numeric values are as follows:
<table>
<tr><th>Measure Value</th><th>Numeric Value</th></tr>
<tr><td>Normal</td><td>0</td></tr>
<tr><td>Warning</td><td>1</td></tr>
<tr><td>Critical</td><td>2</td></tr>
</table>
Note:
By default, this measure reports the Measure Values in the table above to indicate replication health. In the graph of this measure however, the same will be represented using the numeric equivalents only.
</td></tr>
<tr><td>Replication mode</td><td>Indicates the replication mode of this VM.</td><td></td><td>The values that this measure can report and their corresponding numeric values are as follows:
<table>
<tr><th>Measure Value</th><th>Numeric Value</th></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><td>Normal</td><td>0</td></tr>
<tr><td>Primary</td><td>1</td></tr>
<tr><td>Replica</td><td>2</td></tr>
<tr><td>TestReplica</td><td>3</td></tr>
</table>
Note:
By default, this measure reports the Measure Values in the table above to indicate replication mode. In the graph of this measure however, the same will be represented using the numeric equivalents only.
</td></tr>
<tr><td>Last replication time</td><td>Indicates the how much time this VM took to replicate last.</td><td>Secs</td><td></td></tr>
</table>

3.6.9 Hyper-V VM Checkpoints Test

A checkpoint saves the state of each virtual hard disk that is attached to a virtual machine and all of the hard disk's contents, including application data files. For virtual machines on Hyper-V, a checkpoint also saves the hardware configuration information. By creating checkpoints for a virtual machine, you can restore the virtual machine to a previous state.

A typical use of checkpoints is to create a temporary backup before you update the operating system or an application, or make a configuration change on the virtual machine. A checkpoint allows you to restore the virtual machine to its previous state if the operation fails or adversely affects the virtual machine. For virtual machines on Hyper-V, checkpoints are also useful in a test environment where you want to use multiple hardware configurations on a virtual machine.

You can create multiple checkpoints for a virtual machine. However, checkpoints use hard disk space and, when allowed to proliferate, they can affect the performance of a virtual machine when it is running and during such virtual machine operations as migrating a virtual machine or storing it to the library.

To make sure that checkpoints do not affect VM performance, administrators need to continuously track checkpoint growth per VM, identify 'heavy-weight' and obsolete checkpoints that may not be of use any longer, and purge them. The **Hyper-V VM Checkpoints** test helps administrators achieve the same. This test

reports the count of large and aged checkpoints per VM, and reveals the names of thesecheckpoints, so that administrators can decide whether/not these checkpoints can be removed to make more storage space available for the VM.

Note:
This test will report metrics for Microsoft Hyper-V Server 2012 only.

Target of the test:

A Hyper-V / Hyper-V VDI server

Agent executing the test:

An internal agent

Output of the test:

One set of results for each VM on a Hyper-V host monitored

Parameters of the test:

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured.
3. **PORT** - The port number at which the configured HOST listens.
4. **AGE LIMIT IN DAYS** - By default, the value of this parameter is set to 15 days. This implies that the test will report all those snapshots that are more than 15 days old as *Aged snapshots*. If required, you can change the age limit.
5. **SIZE LIMIT IN MB** - By default, the value of this parameter is set to 10000 MB. This implies that the test will report all those snapshots that have a size more than 10000 MB as *Large snapshots*. If required, you can change this limit.
6. **DOMAIN, ADMIN USER, ADMIN PASSWORD,** and **CONFIRM PASSWORD** – By default, this test connects to each virtual guest remotely and attempts to collect “inside view” metrics. Accordingly, the **INSIDE VIEW USING** flag is set to **Remote connection to VM (Windows)** by default. To obtain a remote connection, the test must be configured with the privileges of an administrative user to the domain within which the guests reside. The first step towards this is to specify the **DOMAIN** within which the virtual guests reside. The **ADMIN USER** and **ADMIN PASSWORD** will change according to the domain specification. Discussed below are the different values that the domain parameter can take, and how they impact the **ADMIN USER** and **ADMIN PASSWORD** specifications:

- **If the VMs belong to a single domain:** If the guests belong to a specific domain, then specify the name of that domain against the domain parameter. In this case, any administrative user in that domain will have remote access to all the virtual guests. Therefore, an administrator account in the given domain can be provided in the **ADMIN USER** field and the corresponding password in the **ADMIN PASSWORD** field. Confirm the password by retyping it in the **CONFIRM PASSWORD** text box.
- **If the VMs belong to different domains:** In this case, you might want to provide multiple domain names. If this is done, then, to access the guests in every configured domain, the test should be configured with the required user privileges; this implies that along with multiple **DOMAIN** names, multiple **ADMIN USER** names and **ADMIN PASSWORDS** would also have to be provided. To help administrators provide these user details quickly and easily, the eG administrative interface embeds a special configuration page. To access this page, simply click on the encircled '+' icon that appears alongside the **ADMIN USER** parameter of this test in the test configuration page. To know how to use the special page, refer to Section 3.6.1 of this document.
- **If the INSIDE VIEW USING flag is set to 'eG VM Agent (Windows)'**: On the other hand, if the **INSIDE VIEW USING** flag is set to **eG VM Agent (Windows)** , then it implies that the inside view can be obtained without domain administrator privileges. Therefore, set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.

7. **DD FREQUENCY** - To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test
8. **DETAILED DIAGNOSIS** - To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option.

   The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:

   - The eG manager license should allow the detailed diagnosis capability
   - Both the normal and abnormalfrequencies configured for the detailed diagnosis measures should not be 0.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
Number of checkpoints	Indicates the number of checkpoints for this VM.	Number	
Aged checkpoints count	Indicates the number of checkpoints of this VM, the age of which is more than the **AGE LIMIT** configured for this test.	Number	Use the detailed diagnosis of this measure to identify the aged checkpoints.
Large checkpoints count	Indicates the number of checkpoints of this VM that are of a size greater than the **SIZE LIMIT** configured for this test.	Number	Use the detailed diagnosis of this measure to identify the large-sized checkpoints.

3.7 The Inside View of VMs Layer

The **Outside View of VMs layer** provides an "external" view of the different VM guests – the metrics reported at this layer are based on what the Hyper-V host is seeing about the performance of the individual guests. However, an external view of the VM guest operating system and its applications may not be sufficient. For instance, suppose one of the disk partitions of the guest operating system has reached capacity. This information cannot be gleaned from host operating system. Likewise, bottlenecks such as a longer process run queue or a higher disk queue length are more visible using an internal monitor. Internal monitoring (from within the guest operating system) also provides details about the resource utilization of different application (s) or processes.

The tests mapped to the **Inside View of VMs** layer provide an "internal" view of the workings of each of the guests - these tests execute on an Hyper-V host, but send probes into each of the guest operating systems to analyze how well each guest utilizes the resources that are allocated to it, and how well it handles user sessions, TCP traffic, and network loading.

By default however, clicking on the **Inside View of VMs** layer, does not display the list of tests associated with that layer. Instead, Figure 3.18 appears, which provides you with an overview of individual guest performance (see Figure 3.18).

Figure 3.18: Figure 2.18: A list of guest operating systems on a Hyper-V host and their current state

To return to the layer model of the Hyper-V server and view the tests associated with the **Virtual Servers** layer, click on the **COMPONENT LAYERS** link in Figure 3.18. You can now view the list of tests mapped to the **Inside View of VMs** layer, as depicted by Figure 3.19 below.

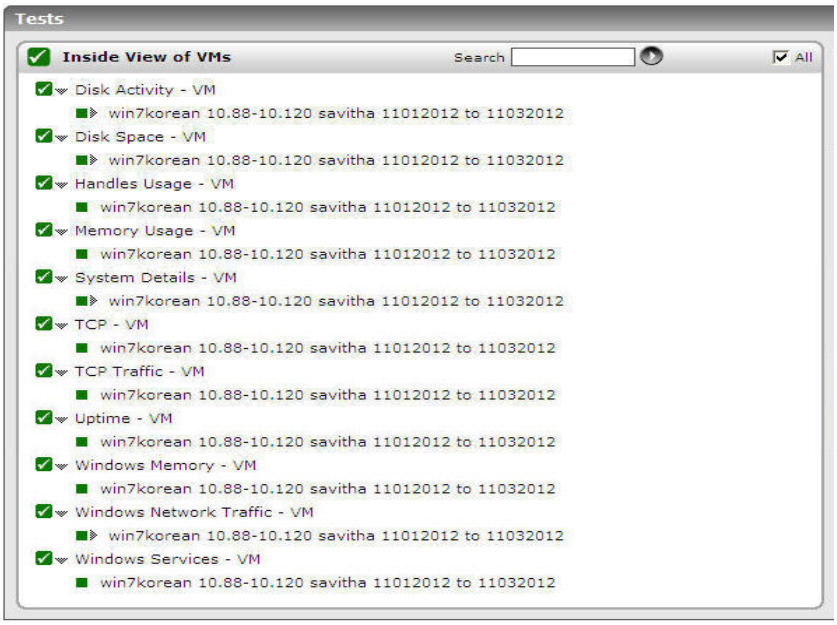


Figure 3.19: The tests mapped to the Inside View of VMs layer

If you want to override this default setting - i.e., if you prefer to view the tests mapped to the **Inside View of VMs**layer first, and then proceed to focus on individual guest performance, follow the steps given below:

- Edit the **eg_ui.ini** file in the **<EG_INSTALL_DIR>\manager\config** directory
- Set the **LAYERMODEL_LINK_TO_VIRTUAL** flag in the file to **false**; this is set to **true** by default.
- Save the **eg_ui.ini** file.

Doing so ensures that as soon as the **Inside View of VMs** layer is clicked, the list of tests mapped to that layer appears, as depicted by Figure 3.20.

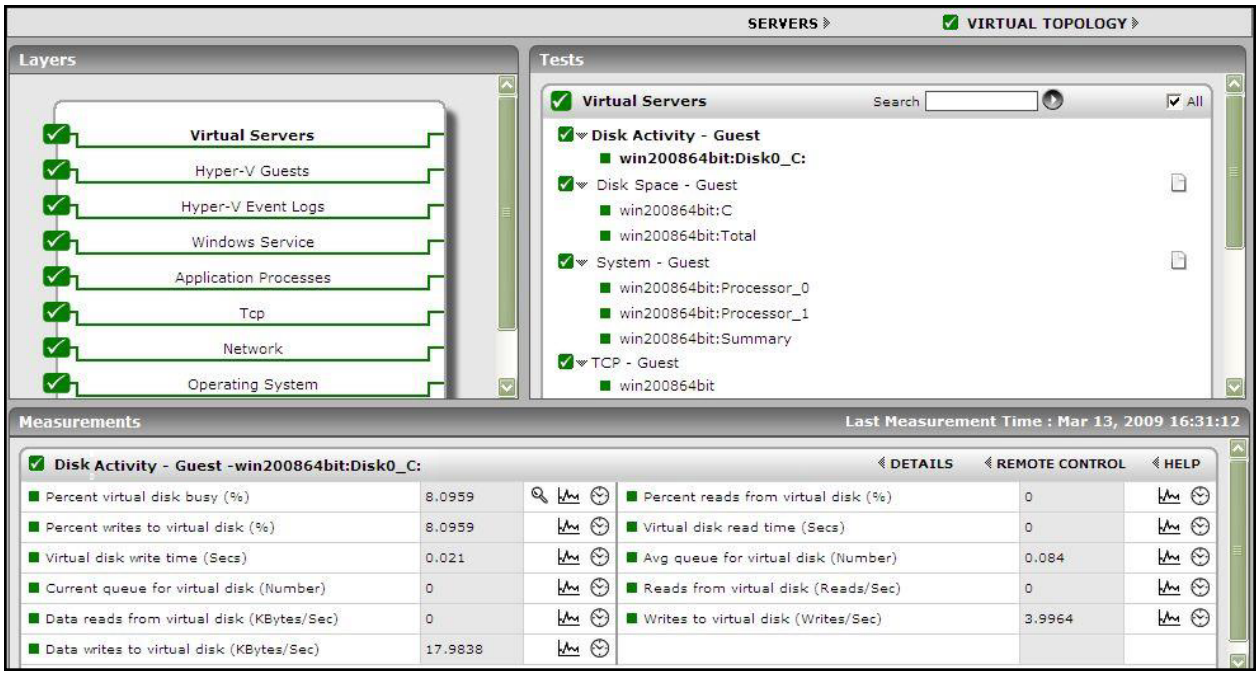


Figure 3.20: The tests mapped to the Virtual Servers layer

If you now want the **Server view** of Figure 3.18, simply click on the **SERVERS** link above the list of tests in Figure 3.20 (indicated by the arrow).

Clicking on any of the guests in the **Server view** leads you to Figure 3.21 that displays all the performance metrics extracted from that guest, in real-time. You are thus enabled to cross-correlate across the various metrics, and quickly detect the root-cause of current/probable disturbances to the internal health of a guest. To view the time-of-day variations in a measure, you can view its graph by clicking on that measure in Figure 3.21.

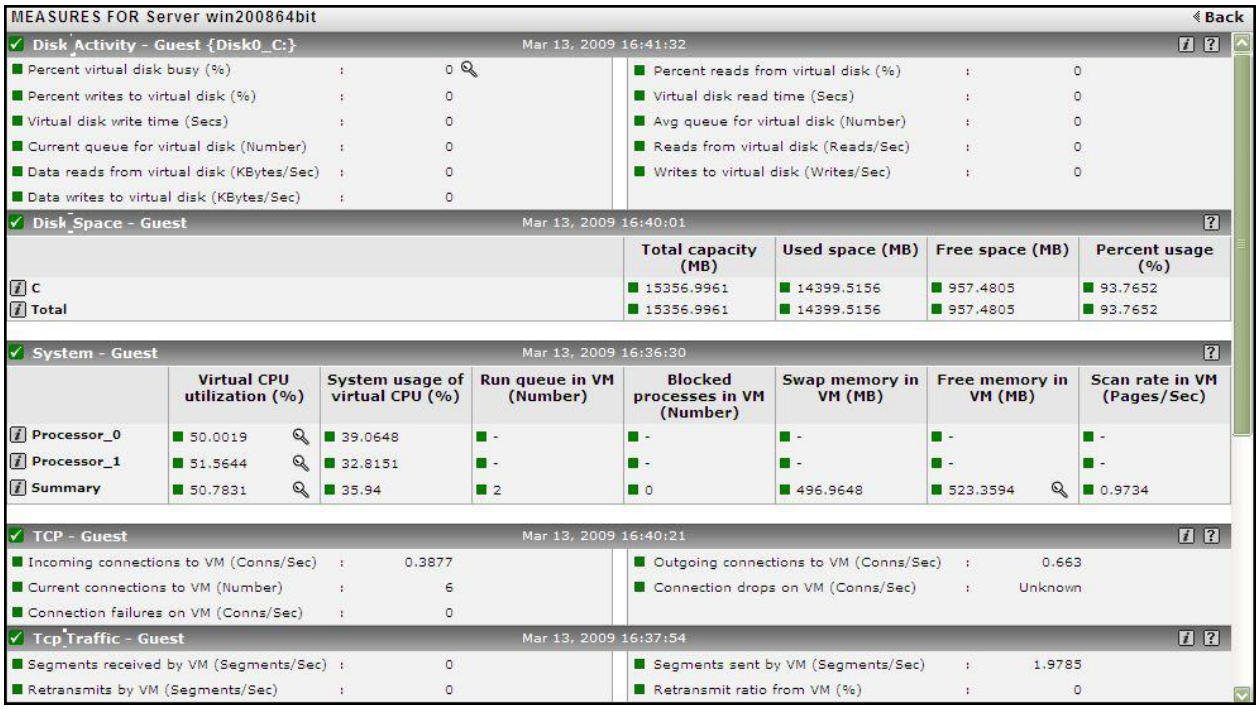


Figure 3.21: Figure 2.30: Measures pertaining to a chosen guest

To view real-time graphs of pre-configured measures (pertaining to the *Hyper-V* host and the guests operating on it), click on the **LIVE GRAPH** link in Figure 3.18. Alternatively, you can click on the icon that appears in the **Tests** panel (see Figure 3.8) when the **Outside View of VMs layer** is clicked. The graph display that appears subsequently (see Figure 3.22) has been organized in such a way that next to every host-pertinent measure graph, the closely related guest-specific measure graph appears. For instance, next to the graph of the 'Cpu utilization' measure of the *Hyper-V Logical Processors* test, you will find a graph of the 'Virtual machine cpu utilization' measure of the *Hyper-V Guests* test. This way, you can easily compare and correlate how well the physical CPU resources are being utilized by both the Hyper-V host and the guests. On the basis of this analysis, you can proactively isolate potential performance issues, and also determine the root-cause of the issue - is it the Hyper-V host? or is it the virtual guest? If you access this page from the **LIVE GRAPH** link in Figure 3.18, then, by default, you will view live graphs pertaining to the *Hyper-V* server. However, you can select a different virtualized component-type and a different virtualized component using the **type** and **ComponentName** lists (respectively) in Figure 3.22.

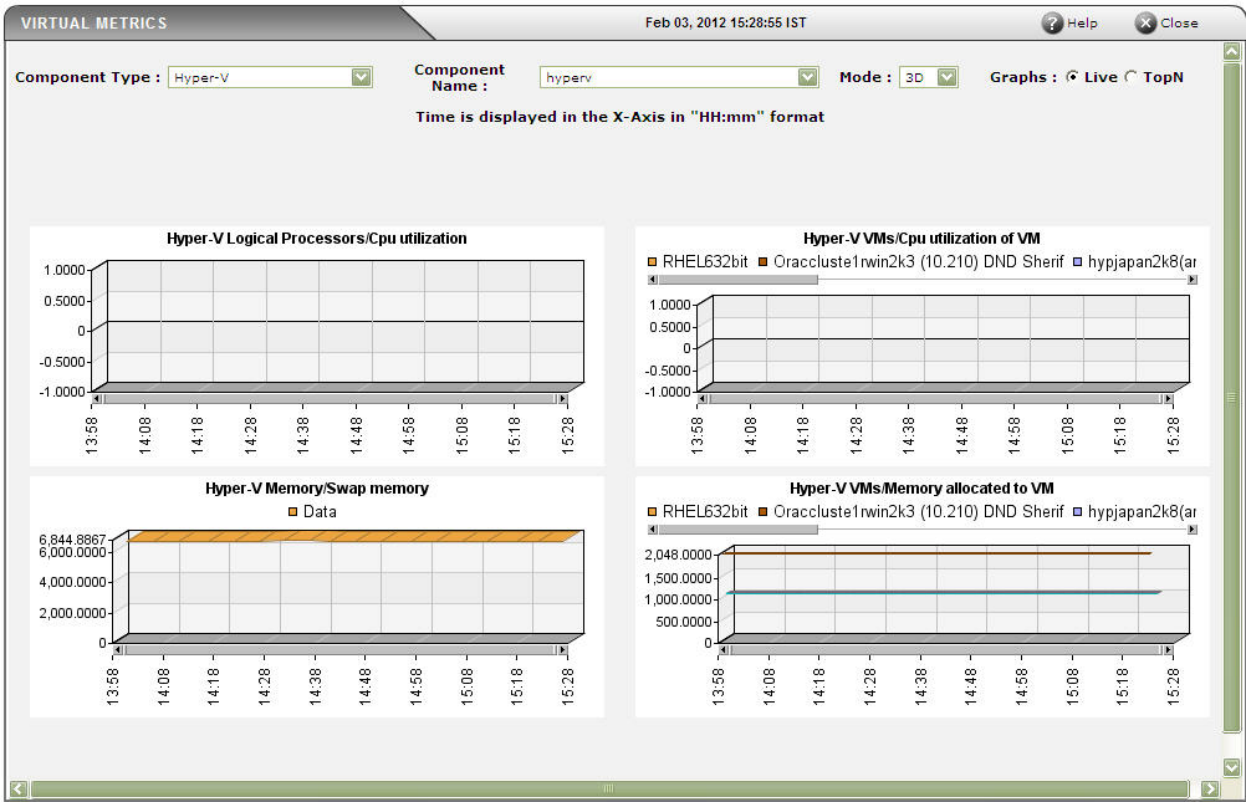


Figure 3.22: Figure 2.31: Live graph comparing physical resource usage of a Hyper-V server (on the left) and resource usage levels of the individual VMs (on the right)

As indicated in Figure 3.19, the tests associated with this layer monitor different aspects of each virtual guest. Disk space utilization, disk activity levels, CPU utilization, memory usage levels, network traffic, etc. are all monitored and reported for each virtual guest hosted on the Hyper-V server. Detailed diagnosis for these tests provide details of individual processes and their utilization levels.

3.7.1 Disk Activity - VM Test

This test reports statistics pertaining to the input/output utilization of each physical disk on a guest.

Target of the test:

A Hyper-V / Hyper-V VDI server

Agent executing the test:

An internal agent

Output of the test:

One set of results for every disk partition on a VM, in the case of a Hyper-V server

On set of results for every disk partition used by a user who is currently logged into a virtual desktop, in the case of a Hyper-V VDI server

First-level descriptor: VM name or username_on_VM

Second-level descriptor: Disk partition

Parameters of the test:

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured.
3. **PORT** - The port at which the **HOST** listens. By default, this is NULL.
4. **INSIDE VIEW USING** - By default, this test communicates with every VM remotely and extracts "inside view" metrics. Therefore, by default, the **INSIDE VIEW USING** flag is set to **Remote connection to VM (Windows)**.

   Typically, to establish this remote connection, eG Enterprise requires that the eG agent be configured with domain administrator privileges. In high-security environments, where the IT staff might have reservations about exposing the credentials of their domain administrators, this approach to extracting "inside view" metrics might not be preferred. In such environments therefore, eG Enterprise provides administrators the option to deploy a piece of software called the **eG VM Agent** on every Windows VM; this VM agent allows the eG agent to collect "inside view" metrics from the Windows VMs **without domain administrator rights.** Refer to Section 1.3.2 for more details on the **eG VM Agent**. To ensure that the "inside view" of Windows VMs is obtained using the eG VM Agent, set the **INSIDE VIEW USING** flag to **eG VM Agent (Windows)**. Once this is done, you can set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.

5. **DOMAIN, ADMIN USER, ADMIN PASSWORD,** and **CONFIRM PASSWORD** – By default, this test connects to each virtual guest remotely and attempts to collect "inside view" metrics. Accordingly, the **INSIDE VIEW USING** flag is set to **Remote connection to VM (Windows)** by default. To obtain a remote connection, the test must be configured with the privileges of an administrative user to the domain within which the guests reside. The first step towards this is to specify the DOMAIN within which the virtual guests reside. The **ADMIN USER** and **ADMIN PASSWORD** will change according to the **DOMAIN** specification. Discussed below are the different values that the **DOMAIN** parameter can take, and how they impact the **ADMIN USER** and **ADMIN PASSWORD** specifications:

   - **If the VMs belong to a single domain:** If the guests belong to a specific domain, then specify the name of that domain against the domain parameter. In this case, any administrative user in that domain will have remote access to all the virtual guests. Therefore, an administrator account in the given domain can be provided in the **ADMIN USER** field and the corresponding password in the **ADMIN PASSWORD** field. Confirm the password by retyping it in the **CONFIRM PASSWORD** text box.
   - **If the VMs belong to different domains :** In this case, you might want to provide multiple

domain names. If this is done, then, to access the guests in every configured domain, the test should be configured with the required user privileges; this implies that along with multiple **DOMAIN** names, multiple **ADMIN USER** names and **ADMIN PASSWORDS** would also have to be provided. To help administrators provide these user details quickly and easily, the eG administrative interface embeds a special configuration page. To access this page, simply click on the encircled '+' icon that appears alongside the **ADMIN USER** parameter of this test in the test configuration page. To know how to use the special page, refer to Section 3.6.1 of this document.

- **If the INSIDE VIEW USING flag is set to 'eG VM Agent (Windows)'**: On the other hand, if the **INSIDE VIEW USING** flag is set to **eG VM Agent (Windows)** , then it implies that the inside view can be obtained without domain administrator privileges. Therefore, set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.

6. **REPORT BY USER** – For the *Microsoft Hyper-V* monitoring model, the **REPORT BY USER** flag is set to **NO** by default, indicating that by default, the guest operating systems on the Hyper-V server are identified using the hostname specified in the operating system. On the other hand, for the *Microsoft Hyper-V VDI* model, this flag is set to **YES** by default; this implies that in case of VDI servers, by default, the guests will be identified using the login of the user who is accessing the guest OS. In other words, in VDI environments, this test will, by default, report measures for every *username_on_virtualmachinename*.
7. **REPORT POWERED OS** - **This flag becomes relevant only if the report by user flag is set to 'Yes'**. If the **REPORT POWERED OS** flag is set to **Yes** (which is the default setting), then this test will report measures for even those VMs that do not have any users logged in currently. Such guests will be identified by their *virtualmachine name* and not by the *username_on_virtualmachinename*. On the other hand, if the **REPORT POWERED OS** flag is set to **No**, then this test will not report measures for those VMs to which no users are logged in currently.

   Note:

   While performing VM discovery, the eG agent will not discover the operating system of the VMs configured in the **IGNORE VMS INSIDE VIEW** text box.

8. **IGNORE VMS INSIDE VIEW** - Administrators of some high security Hyper-V environments might not have permissions to internally monitor one/more VMs. The eG agent can be configured to **not obtain the 'inside view' of such 'inaccessible' VMs** using the **IGNORE VMS INSIDE VIEW** parameter. Against this parameter, you can provide a comma-separated list of VM names, or VM name patterns, for which the inside view need not be obtained. For instance, your **IGNORE VMS INSIDE VIEW** specification can be: **xp,*lin*,win*,vista*. Here, the * (asterisk) is used to denote leading and trailing spaces (as the case may be). By default, this parameter is set to *none* indicating that the eG agent obtains the inside view of all VMs on a Hyper-V host by default.
9. **EXCLUDE VMS** - Administrators of some virtualized environments may not want to monitor some of their less-critical VMs - for instance, VM templates - both from 'outside' and from 'inside'. The eG agent in this case can be configured to completely exclude such VMs from its monitoring purview. To achieve this, provide a comma-separated list of VMs to be excluded from monitoring in the **EXCLUDE VMS** text

box. Instead of VMs, VM name patterns can also be provided here in a comma-separated list. For example, your **EXCLUDE VMS** specification can be: **xp,*lin*,win*,vista*. Here, the * (asterisk) is used to denote leading and trailing spaces (as the case may be). By default, this parameter is set to *none* indicating that the eG agent obtains the inside and outside views of all VMs on a virtual host by default. By providing a comma-separated list of VMs/VM name patterns in the **EXCLUDE VMS** text box, you can make sure the eG agent stops collecting 'inside' and 'outside' view metrics for a configured set of VMs.

10. **IGNORE WINNT** – By default, the eG agent does not support the *inside view* for VMs executing on **Windows NT** operating systems. Accordingly, the **IGNORE WINNT** flag is set to **Yes** by default.
11. **DETAILED DIAGNOSIS** - To make diagnosis more efficient and accurate, the eG suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option.

    The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:

    - The eG manager license should allow the detailed diagnosis capability
    - Both the normal and abnormalfrequencies configured for the detailed diagnosis measures should not be 0.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
Percent virtual disk busy	Indicates the percentage of elapsed time during which the disk is busy processing requests (i.e., reads or writes).	Percent	Comparing the percentage of time that the different disks are busy, an administrator can determine whether load is properly balanced across the different disks.
Percent reads from virtual disk	Indicates the p*ercentage of elapsed time that the selected disk drive is*	Percent	

Measurement	Description	Measurement Unit	Interpretation
	busy servicing read requests.		
Percent writes to virtual disk	Indicates the p*ercentage of elapsed time that the selected disk drive is busy servicing write requests*.	Percent	
Virtual disk read time	Indicates the average time in seconds of a read of data from the disk.	Secs	
Virtual disk write time	Indicates the average time in seconds of a write of data from the disk.	Secs	
Avg. queue for virtual disk	Indicates the average number of both read and write requests that were queued for the selected disk during the sample interval.	Number	
Current queue for virtual disk	The number of requests outstanding on the disk at the time the performance data is collected.	Number	This measure includes requests in service at the time of the snapshot. This is an instantaneous length, not an average over the time interval. Multi-spindle disk devices can have multiple requests active at one time, but other concurrent requests are awaiting service. This counter might reflect a transitory high or low queue length, but if there is a sustained load on the disk drive, it is likely that this will be consistently high. Requests experience delays proportional to the length of this queue minus the number of spindles on the disks. This difference should average less than two for good performance.

Measurement	Description	Measurement Unit	Interpretation
Reads from virtual disk	Indicates the number of reads happening on a logical disk per second.	Reads/Sec	A dramatic increase in this value may be indicative of an I/O bottleneck on the guest.
Data reads from virtual disk	Indicates the rate at which bytes are transferred from the disk during read operations.	KB/Sec	A very high value indicates an I/O bottleneck on the guest.
Writes to virtual disk	Indicates the number of writes happening on a local disk per second.	Writes/Sec	A dramatic increase in this value may be indicative of an I/O bottleneck on the guest.
Data writes to virtual disk	Indicates the rate at which bytes are transferred from the disk during write operations.	KB/Sec	A very high value indicates an I/O bottleneck on the guest.
Disk service time	Indicates the average time that this disk took to service each transfer request (i.e., the average I/O operation time)	Secs	A sudden rise in the value of this measure can be attributed to a large amount of information being input or output. A consistent increase however, could indicate an I/O processing bottleneck.
Disk queue time	Indicates the average time that transfer requests waited idly on queue for this disk.	Secs	Ideally, the value of this measure should be low.
Disk I/O time	Indicates the average time taken for read and write operations of this disk.	Secs	The value of this measure is the sum of the values of the Disk service time and Disk queue time measures. A consistent increase in the value of this measure could indicate a latency in I/O processing.

The detailed diagnosis of the *Percent virtual disk busy* measure, if enabled, provides information such as the Process IDs executing on the disk, the Process names, the rate at which I/O read and write requests were issued by each of the processes, and the rate at which data was read from and written into the disk by each of

the processes. In the event of excessive disk activity, the details provided in the detailed diagnosis page will enable users to figure out which process is performing the I/O operation that is keeping the disk busy.

Shows the IO operations done by the processes

Time	ID Process	ProcessName	IO Rate (Bytes/sec)	IO Read Rate (Bytes/sec)	IO Read Ops Rate (Ops/Sec)	IO Write Rate (Bytes/sec)	IO Write Ops Rate (Ops/sec)
Mar 13, 2009 15:55:36							
	384	svchost#4	40504.75	40414.24	5.9	90.52	0.98
	3004	vmwp#1	22573.04	8694.96	46.57	13878.09	46.57
	696	lsass	5098.51	2823.76	30.5	2274.75	29.52
	2348	vmms	4005.74	1711.97	24.6	2293.77	24.6
	1616	js	2919.2	1770.67	17.05	1148.53	7.54
	3012	vmwp#2	86.58	43.29	0.98	43.29	0.98
	1020	svchost#2	10.49	10.49	1.31	0	0
	4	System	10.49	0	0	10.49	1.31

Figure 3.23: The detailed diagnosis of the Percent virtual disk busy measure

3.7.2 Disk Space - VM Test

This test monitors the space usage of every disk partition on a guest.

Target of the test:

A Hyper-V / Hyper-V VDI server

Agent executing the test:

An internal agent

Output of the test:

One set of results for every disk partition on a VM, in the case of a Hyper-V server

On set of results for every disk partition used by a user who is currently logged into a virtual desktop, in the case of a Hyper-V VDI server

First-level descriptor: VM name or username_on_VM

Second-level descriptor: Disk partition

Parameters of the test:

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured.
3. **PORT** - The port at which the **HOST** listens. By default, this is NULL.
4. **INSIDE VIEW USING** - By default, this test communicates with every VM remotely and extracts “inside

view" metrics. Therefore, by default, the **INSIDE VIEW USING** flag is set to **Remote connection to VM (Windows)**.

Typically, to establish this remote connection, eG Enterprise requires that the eG agent be configured with domain administrator privileges. In high-security environments, where the IT staff might have reservations about exposing the credentials of their domain administrators, this approach to extracting "inside view" metrics might not be preferred. In such environments therefore, eG Enterprise provides administrators the option to deploy a piece of software called the **eG VM Agent** on every Windows VM; this VM agent allows the eG agent to collect "inside view" metrics from the Windows VMs **without domain administrator rights.** Refer to Section 1.3.2 for more details on the **eG VM Agent**. To ensure that the "inside view" of Windows VMs is obtained using the eG VM Agent, set the **INSIDE VIEW USING** flag to **eG VM Agent (Windows)**. Once this is done, you can set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD**parameters to *none*.

5. **DOMAIN, ADMIN USER, ADMIN PASSWORD,**and **CONFIRM PASSWORD**– By default, this test connects to each virtual guest remotely and attempts to collect "inside view" metrics. Accordingly, the **INSIDE VIEW USING** flag is set to **Remote connection to VM (Windows)** by default. To obtain a remote connection, the test must be configured with the privileges of an administrative user to the domain within which the guests reside. The first step towards this is to specify the DOMAIN within which the virtual guests reside. The **ADMIN USER** and **ADMIN PASSWORD** will change according to the **DOMAIN** specification. Discussed below are the different values that the **DOMAIN** parameter can take, and how they impact the **ADMIN USER** and **ADMIN PASSWORD** specifications:

   - **If the VMs belong to a single domain:** If the guests belong to a specific domain, then specify the name of that domain against the domain parameter. In this case, any administrative user in that domain will have remote access to all the virtual guests. Therefore, an administrator account in the given domain can be provided in the **ADMIN USER** field and the corresponding password in the **ADMIN PASSWORD** field. Confirm the password by retyping it in the **CONFIRM PASSWORD** text box.
   - **If the VMs belong to different domains :** In this case, you might want to provide multiple domain names. If this is done, then, to access the guests in every configured domain, the test should be configured with the required user privileges; this implies that along with multiple **DOMAIN** names, multiple **ADMIN USER** names and **ADMIN PASSWORDS** would also have to be provided. To help administrators provide these user details quickly and easily, the eG administrative interface embeds a special configuration page. To access this page, simply click on the encircled '+' icon that appears alongside the **ADMIN USER** parameter of this test in the test configuration page. To know how to use the special page, refer to Section 3.6.1 of this document.
   - **If the INSIDE VIEW USING flag is set to 'eG VM Agent (Windows)'**: On the other hand, if the **INSIDE VIEW USING** flag is set to **eG VM Agent (Windows)** , then it implies that the inside view can be obtained without domain administrator privileges. Therefore, set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.

6. **REPORT BY USER** – For the *Microsoft Hyper-V* monitoring model, the **REPORT BY USER** flag is set to **NO** by default, indicating that by default, the guest operating systems on the Hyper-V server are identified using the hostname specified in the operating system. On the other hand, for the *Microsoft Hyper-V VDI* model, this flag is set to **YES** by default; this implies that in case of VDI servers, by default, the guests will be identified using the login of the user who is accessing the guest OS. In other words, in VDI environments, this test will, by default, report measures for every *username_on_virtualmachinename*.
7. **REPORT POWERED OS** - **This flag becomes relevant only if the report by user flag is set to 'Yes'**. If the **REPORT POWERED OS** flag is set to **Yes** (which is the default setting), then this test will report measures for even those VMs that do not have any users logged in currently. Such guests will be identified by their *virtualmachine name* and not by the *username_on_virtualmachinename*. On the other hand, if the **REPORT POWERED OS** flag is set to **No**, then this test will not report measures for those VMs to which no users are logged in currently.

   Note:

   While performing VM discovery, the eG agent will not discover the operating system of the VMs configured in the **IGNORE VMS INSIDE VIEW** text box.

8. **IGNORE VMS INSIDE VIEW** - Administrators of some high security Hyper-V environments might not have permissions to internally monitor one/more VMs. The eG agent can be configured to **not obtain the 'inside view' of such 'inaccessible' VMs** using the **IGNORE VMS INSIDE VIEW** parameter. Against this parameter, you can provide a comma-separated list of VM names, or VM name patterns, for which the inside view need not be obtained. For instance, your **IGNORE VMS INSIDE VIEW** specification can be: **xp,*lin*,win*,vista*. Here, the * (asterisk) is used to denote leading and trailing spaces (as the case may be). By default, this parameter is set to *none* indicating that the eG agent obtains the inside view of all VMs on a Hyper-V host by default.
9. **EXCLUDE VMS** - Administrators of some virtualized environments may not want to monitor some of their less-critical VMs - for instance, VM templates - both from 'outside' and from 'inside'. The eG agent in this case can be configured to completely exclude such VMs from its monitoring purview. To achieve this, provide a comma-separated list of VMs to be excluded from monitoring in the **EXCLUDE VMS** text box. Instead of VMs, VM name patterns can also be provided here in a comma-separated list. For example, your **EXCLUDE VMS** specification can be: **xp,*lin*,win*,vista*. Here, the * (asterisk) is used to denote leading and trailing spaces (as the case may be). By default, this parameter is set to *none* indicating that the eG agent obtains the inside and outside views of all VMs on a virtual host by default. By providing a comma-separated list of VMs/VM name patterns in the **EXCLUDE VMS** text box, you can make sure the eG agent stops collecting 'inside' and 'outside' view metrics for a configured set of VMs.
10. **IGNORE WINNT** – By default, the eG agent does not support the *inside view* for VMs executing on **Windows NT** operating systems. Accordingly, the **IGNORE WINNT** flag is set to **Yes** by default.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
Total capacity	Indicates the total capacity of a disk partition; for the **Total** descriptor, this measure reports the sum of the total capacity of all disk partitions.	MB	
Used space	Indicates the amount of space used in a disk partition; for the **Total** descriptor, this measure reports the sum of space used across all disk partitions.	MB	
Free space	Indicates the *current free space available for each disk partition of a system*; for the **Total** descriptor, this measure reports the sum of the unused space in all disk partitions.	MB	
Percent usage	Indicates the *percentage of space usage on each disk partition of a system; for the* **Total** descriptor, this measure reports the percentage of disk space used across all disk partitions.	Percent	A value close to 100% can indicate a potential problem situation where applications executing on the guest may not be able to write data to the disk partition (s) with very high usage.

3.7.3 System Details - VM Test

This test collects various metrics pertaining to the CPU and memory usage of every processor supported by a guest. The details of this test are as follows:

Target of the test:

A Hyper-V / Hyper-V VDI server

Agent executing the test:

An internal agent

Output of the test:

For a Hyper-V server, one set of results for every processor on each VM

For a Hyper-V VDI server, one set of results for every processor used by the user who is currently logged into each VM

First-level descriptor: VM name or User on VM

Second-level descriptor: Processor name

Parameters of the test:

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured.
3. **PORT** - The port at which the **HOST** listens. By default, this is NULL.
4. **INSIDE VIEW USING** - By default, this test communicates with every VM remotely and extracts "inside view" metrics. Therefore, by default, the **INSIDE VIEW USING** flag is set to **Remote connection to VM (Windows)**.

   Typically, to establish this remote connection, eG Enterprise requires that the eG agent be configured with domain administrator privileges. In high-security environments, where the IT staff might have reservations about exposing the credentials of their domain administrators, this approach to extracting "inside view" metrics might not be preferred. In such environments therefore, eG Enterprise provides administrators the option to deploy a piece of software called the **eG VM Agent** on every Windows VM; this VM agent allows the eG agent to collect "inside view" metrics from the Windows VMs **without domain administrator rights.** Refer to Section 1.3.2 for more details on the **eG VM Agent**. To ensure that the "inside view" of Windows VMs is obtained using the eG VM Agent, set the **INSIDE VIEW USING** flag to **eG VM Agent (Windows)**. Once this is done, you can set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.
5. **DOMAIN, ADMIN USER, ADMIN PASSWORD,** and **CONFIRM PASSWORD** – By default, this test connects to each virtual guest remotely and attempts to collect "inside view" metrics. Accordingly, the **INSIDE VIEW USING** flag is set to **Remote connection to VM (Windows)** by default. To obtain a remote connection, the test must be configured with the privileges of an administrative user to the domain within which the guests reside. The first step towards this is to specify the DOMAIN within which the virtual guests reside. The **ADMIN USER** and **ADMIN PASSWORD** will change according to the **DOMAIN** specification. Discussed below are the different values that the **DOMAIN** parameter can take, and how they impact the **ADMIN USER** and **ADMIN PASSWORD** specifications:
   - **If the VMs belong to a single domain:** If the guests belong to a specific domain, then specify the name of that domain against the domain parameter. In this case, any administrative user in that domain will have remote access to all the virtual guests. Therefore, an administrator account in the given domain can be provided in the **ADMIN USER** field and the corresponding password in the **ADMIN PASSWORD** field. Confirm the password by retyping it in the **CONFIRM PASSWORD** text box.

- I**f the VMs belong to different domains:** In this case, you might want to provide multiple domain names. If this is done, then, to access the guests in every configured domain, the test should be configured with the required user privileges; this implies that along with multiple **DOMAIN** names, multiple **ADMIN USER** names and **ADMIN PASSWORDS** would also have to be provided. To help administrators provide these user details quickly and easily, the eG administrative interface embeds a special configuration page. To access this page, simply click on the encircled '+' icon that appears alongside the . To know how to use the special page, refer to Section 3.6.1 of this document.
- **If the INSIDE VIEW USING flag is set to 'eG VM Agent (Windows)'**: On the other hand, if the **INSIDE VIEW USING** flag is set to **eG VM Agent (Windows)**, then it implies that the inside view can be obtained without domain administrator privileges. Therefore, set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.

4. **REPORT BY USER** – For the *Hyper-V* monitoring model, the REPORT BY USER flag is set to **NO** by default, indicating that by default, the guest operating systems on the Hyper-V server are identified using the hostname specified in the operating system. On the other hand, for the *Hyper-V VDI* model, this flag is set to **YES** by default; this implies that in case of VDI servers, by default, the guests will be identified using the login of the user who is accessing the guest OS. In other words, in VDI environments, this test will, by default, report measures for every *username_on_virtualmachinename*.
5. **REPORT POWERED OS** - **This flag becomes relevant only if the REPORT BY USER flag is set to 'Yes'.**

   If the **REPORT POWERED OS** flag is set to **Yes** (which is the default setting), then this test will report measures for even those VMs that do not have any users logged in currently. Such guests will be identified by their *virtualmachine name* and not by the *username_on_virtualmachinename*. On the other hand, if the **REPORT POWERED OS** flag is set to **No**, then this test will not report measures for those VMs to which no users are logged in currently.
6. **IGNORE VMS INSIDE VIEW** - Administrators of some high security Hyper-V environments might not have permissions to internally monitor one/more VMs. The eG agent can be configured to **not obtain the 'inside view' of such 'inaccessible' VMs** using the **IGNORE VMS INSIDE VIEW** parameter. Against this parameter, you can provide a comma-separated list of VM names, or VM name patterns, for which the inside view need not be obtained. For instance, your **IGNORE VMS INSIDE VIEW** specification can be: **xp,*lin*,win*,vista*. Here, the * (asterisk) is used to denote leading and trailing spaces (as the case may be). By default, this parameter is set to *none* indicating that the eG agent obtains the inside view of all VMs on a Hyper-V host by default.

   Note:
   While performing VM discovery, the eG agent will not discover the operating system of the VMs configured in the **IGNORE VMS INSIDE VIEW** text box.
7. **EXCLUDE VMS** - Administrators of some virtualized environments may not want to monitor some of their less-critical VMs - for instance, VM templates - both from 'outside' and from 'inside'. The eG agent in this case can be configured to completely exclude such VMs from its monitoring purview. To achieve this, provide a comma-separated list of VMs to be excluded from monitoring in the **EXCLUDE VMS** text box. Instead of VMs, VM name patterns can also be provided here in a comma-separated list. For

example, your **EXCLUDE VMS** specification can be: **xp,*lin*,win*,vista*. Here, the * (asterisk) is used to denote leading and trailing spaces (as the case may be). By default, this parameter is set to *none* indicating that the eG agent obtains the inside and outside views of all VMs on a virtual host by default. By providing a comma-separated list of VMs/VM name patterns in the **EXCLUDE VMS** text box, you can make sure the eG agent stops collecting 'inside' and 'outside' view metrics for a configured set of VMs.

8. **IGNORE WINNT** – By default, the eG agent does not support the *inside view* for VMs executing on **Windows NT** operating systems. Accordingly, the **IGNORE WINNT** flag is set to **Yes** by default.
9. **ENABLE MEMORY DIAGNOSIS** - By default, the ENABLE MEMORY DIAGNOSIS flag is set to **NO**, indicating that detailed diagnosis will not be available for the *Free memory in VM* measure reported by this test by default. If you want to view the detailed diagnosis of the *Free memory in VM* measure - i.e., to view the top 10 processes on the target VM that are utilizing memory excessively - you can change this flag to **YES**.
10. **DETAILED DIAGNOSIS** - To make diagnosis more efficient and accurate, the eG suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option.

    The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:

    - The eG manager license should allow the detailed diagnosis capability
    - Both the normal and abnormalfrequencies configured for the detailed diagnosis measures should not be 0.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
Virtual CPU utilization	This measurement indicates the percentage of CPU utilized by the processor.	Percent	A high value could signify a CPU bottleneck. The CPU utilization may be high because a few processes are consuming a lot of CPU, or because there are too many processes contending for a limited resource. The detailed diagnosis of this test reveals the top-10 CPU-intensive processes on the guest.
System usage of virtual CPU	Indicates the percentage of	Percent	An unusually high value indicates a

Measurement	Description	Measurement Unit	Interpretation
	CPU time spent for system-level processing.		problem and may be due to too many system- level tasks executing simultaneously.
Run queue in VM	Indicates the instantaneous length of the queue in which threads are waiting for the processor cycle. This length does not include the threads that are currently being executed.	Number	A value consistently greater than 2 indicates that many processes could be simultaneously contending for the processor.
Blocked processes in VM	Indicates the number of processes blocked for I/O, paging, etc.	Number	A high value could indicate an I/O problem on the guest (e.g., a slow disk).
Swap memory in VM	Denotes the committed amount of virtual memory. This corresponds to the space reserved for virtual memory on disk paging file(s).	MB	An unusually high value for the swap usage can indicate a memory bottleneck. Check the memory utilization of individual processes to figure out the process (es) that has (have) maximum memory consumption and look to tune their memory usages and allocations accordingly.
Free memory in VM	Indicates the free memory available.	MB	This measure typically indicates the amount of memory available for use by applications running on the target VM. On Unix operating systems (AIX and Linux), the operating system tends to use parts of the available memory for caching files, objects, etc. When applications require additional memory, this is released from the operating system cache. Hence, to understand the true free memory that

Measurement	Description	Measurement Unit	Interpretation
			is available to applications, the eG agent reports the sum of the free physical memory and the operating system cache memory size as the value of the *Free memory in VM* measure while monitoring AIX and Linux guest operating systems. The detailed diagnosis of this measure, if enabled, lists the top 10 processes responsible for maximum memory consumption on the target VM.
Scan rate in VM	Indicates the memory scan rate.	Pages/Sec	A high value is indicative of memory thrashing. Excessive thrashing can be detrimental to guest performance.

Note:

For multi-processor systems, where the CPU statistics are reported for each processor on the system, the statistics that are system-specific (e.g., run queue length, free memory, etc.) are only reported for the "Summary" descriptor of this test.

The detailed diagnosis capability of the *Virtual CPU utilization* measure, if enabled, provides a listing of the top 10 CPU-consuming processes (see 3.7.3). In the event of a Cpu bottleneck, this information will enable users to identify the processes consuming a high percentage of CPU time. The users may then decide to stop such processes, so as to release the CPU resource for more important processing purposes.

Lists the top 10 CPU processes

Time	PID	%CPU	ARGS
Mar 13, 2009 16:32:06			
	2452	12.32	java
Mar 13, 2009 16:21:42			
	500	7.04	csrss
	876	1.41	svchost
	1576	1.41	js
Mar 13, 2009 16:11:59			
	2300	4.27	java
	1344	1.42	vmicsvc
	500	1.42	csrss
	2828	1.42	vmggetcpu

Figure 3.24: The top 10 CPU consuming processes

Note:

While instantaneous spikes in CPU utilization are captured by the eG agents and displayed in the Measures page, the detailed diagnosis will not capture/display such instantaneous spikes. Instead, detailed diagnosis will display only a consistent increase in CPU utilization observed over a period of time.

The detailed diagnosis of the *Free memory in VM* measure, if enabled, lists the top 10 processes responsible for maximum memory consumption on the guest (see Figure 3.25). This information will enable administrators to identify the processes that are causing the depletion in the amount of free memory on the host. The administrators can then decide to kill such expensive processes.

Lists the top 10 memory consuming processes

Time	PID	Memory used(MB)	ARGS
Mar 13, 2009 16:32:06			
	1576	79.36	js
	1020	31.99	svchost
	2452	21.11	java
	548	20.11	svchost
	948	13.89	logonui
	648	12.84	lsass
	1272	11.48	spoolsv
	1092	11.29	svchost
	344	10.98	svchost
	244	10.61	slsvc

Figure 3.25: The detailed diagnosis of the Free memory in VM measure listing the top 10 memory consuming processes

3.7.4 Uptime - VM Test

In most virtualized environments, it is essential to monitor the uptime of VMs hosting critical server applications in the infrastructure. By tracking the uptime of each of the VMs, administrators can determine what percentage of time a VM has been up. Comparing this value with service level targets, administrators can determine the most trouble-prone areas of the virtualized infrastructure.

In some environments, administrators may schedule periodic reboots of their VM. By knowing that a specific VM has been up for an unusually long time, an administrator may come to know that the scheduled reboot task is not working on a VM.

The Uptime - VM test included in the eG agent monitors the uptime of each VM on a Hyper-V server.

Target of the test:

A Hyper-V / Hyper-V VDI server

Agent executing the test:

An internal agent

Output of the test:

One set of results for every guest on the Hyper-V server

Parameters of the test:

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured.
3. **PORT** - The port at which the **HOST** listens. By default, this is NULL.
4. **INSIDE VIEW USING** - By default, this test communicates with every VM remotely and extracts "inside view" metrics. Therefore, by default, the **INSIDE VIEW USING** flag is set to **Remote connection to VM (Windows)**.

   Typically, to establish this remote connection, eG Enterprise requires that the eG agent be configured with domain administrator privileges. In high-security environments, where the IT staff might have reservations about exposing the credentials of their domain administrators, this approach to extracting "inside view" metrics might not be preferred. In such environments therefore, eG Enterprise provides administrators the option to deploy a piece of software called the **eG VM Agent** on every Windows VM; this VM agent allows the eG agent to collect "inside view" metrics from the Windows VMs **without domain administrator rights.** Refer to Section 1.3.2 for more details on the **eG VM Agent**. To ensure that the "inside view" of Windows VMs is obtained using the eG VM Agent, set the **INSIDE VIEW USING** flag to **eG VM Agent (Windows)**. Once this is done, you can set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.
5. **DOMAIN, ADMIN USER, ADMIN PASSWORD,** and **CONFIRM PASSWORD** – By default, this test connects to each virtual guest remotely and attempts to collect "inside view" metrics. Accordingly, the **INSIDE VIEW USING** flag is set to **Remote connection to VM (Windows)** by default. To obtain a remote connection, the test must be configured with the privileges of an administrative user to the domain within which the guests reside. The first step towards this is to specify the DOMAIN within which the virtual guests reside. The **ADMIN USER** and **ADMIN PASSWORD** will change according to the **DOMAIN** specification. Discussed below are the different values that the **DOMAIN** parameter can take, and how they impact the **ADMIN USER** and **ADMIN PASSWORD** specifications:
   - **If the VMs belong to a single domain:** If the guests belong to a specific domain, then specify the name of that domain against the domain parameter. In this case, any administrative user in that domain will have remote access to all the virtual guests. Therefore, an administrator account in the given domain can be provided in the **ADMIN USER** field and the corresponding password in the **ADMIN PASSWORD** field. Confirm the password by retyping it in the **CONFIRM PASSWORD** text box.
   - **If the VMs belong to different domains:**In this case, you might want to provide multiple domain names. If this is done, then, to access the guests in every configured domain, the test should be configured with the required user privileges; this implies that along with multiple **DOMAIN** names, multiple **ADMIN USER** names and **ADMIN PASSWORDS** would also have to be provided. To help administrators provide these user details quickly and easily, the eG administrative interface embeds a special configuration page. To access this page, simply click on the encircled '+' icon that appears alongside the . To know how to use the special page, refer to Section 3.6.1 of this document.

- **If the INSIDE VIEW USING flag is set to 'eG VM Agent (Windows)'**: On the other hand, if the **INSIDE VIEW USING** flag is set to **eG VM Agent (Windows)**, then it implies that the inside view can be obtained without domain administrator privileges. Therefore, set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.

4. **REPORT BY USER** – For the *Hyper-V* monitoring model, the REPORT BY USER flag is set to **NO** by default, indicating that by default, the guest operating systems on the Hyper-V server are identified using the hostname specified in the operating system. On the other hand, for the *Hyper-V VDI* model, this flag is set to **YES** by default; this implies that in case of VDI servers, by default, the guests will be identified using the login of the user who is accessing the guest OS. In other words, in VDI environments, this test will, by default, report measures for every *username_on_virtualmachinename*.
5. **REPORT POWERED OS** - **This flag becomes relevant only if the REPORT BY USER flag is set to 'Yes'.**

   If the **REPORT POWERED OS** flag is set to **Yes** (which is the default setting), then this test will report measures for even those VMs that do not have any users logged in currently. Such guests will be identified by their *virtualmachine name* and not by the *username_on_virtualmachinename*. On the other hand, if the **REPORT POWERED OS** flag is set to **No**, then this test will not report measures for those VMs to which no users are logged in currently.
6. **IGNORE VMS INSIDE VIEW** - Administrators of some high security Hyper-V environments might not have permissions to internally monitor one/more VMs. The eG agent can be configured to **not obtain the 'inside view' of such 'inaccessible' VMs** using the **IGNORE VMS INSIDE VIEW** parameter. Against this parameter, you can provide a comma-separated list of VM names, or VM name patterns, for which the inside view need not be obtained. For instance, your **IGNORE VMS INSIDE VIEW** specification can be: **xp,*lin*,win*,vista*. Here, the * (asterisk) is used to denote leading and trailing spaces (as the case may be). By default, this parameter is set to *none* indicating that the eG agent obtains the inside view of all VMs on a Hyper-V host by default.

   Note:
   While performing VM discovery, the eG agent will not discover the operating system of the VMs configured in the **IGNORE VMS INSIDE VIEW** text box.
7. **EXCLUDE VMS** - Administrators of some virtualized environments may not want to monitor some of their less-critical VMs - for instance, VM templates - both from 'outside' and from 'inside'. The eG agent in this case can be configured to completely exclude such VMs from its monitoring purview. To achieve this, provide a comma-separated list of VMs to be excluded from monitoring in the **EXCLUDE VMS** text box. Instead of VMs, VM name patterns can also be provided here in a comma-separated list. For example, your **EXCLUDE VMS** specification can be: **xp,*lin*,win*,vista*. Here, the * (asterisk) is used to denote leading and trailing spaces (as the case may be). By default, this parameter is set to *none* indicating that the eG agent obtains the inside and outside views of all VMs on a virtual host by default. By providing a comma-separated list of VMs/VM name patterns in the **EXCLUDE VMS** text box, you can make sure the eG agent stops collecting 'inside' and 'outside' view metrics for a configured set of VMs.
8. **IGNORE WINNT** – By default, the eG agent does not support the *inside view* for VMs executing on **Windows NT** operating systems. Accordingly, the **IGNORE WINNT** flag is set to **Yes** by default.

9. **DETAILED DIAGNOSIS** - To make diagnosis more efficient and accurate, the eG suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option.

   The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:

   - The eG manager license should allow the detailed diagnosis capability
   - Both the normal and abnormalfrequencies configured for the detailed diagnosis measures should not be 0.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
Has the VM been rebooted?	Indicates whether the VM has been rebooted during the last measurement period or not.	Boolean	If this measure shows 1, it means that the guest was rebooted during the last measurement period. By checking the time periods when this metric changes from 0 to 1, an administrator can determine the times when this guest was rebooted.
Uptime of the VM during the last measure period	Indicates the time period that the VM has been up since the last time this test ran.	Secs	If the guest has not been rebooted during the last measurement period and the agent has been running continuously, this value will be equal to the measurement period. If the

Measurement	Description	Measurement Unit	Interpretation
			guest was rebooted during the last measurement period, this value will be less than the measurement period of the test. For example, if the measurement period is 300 secs, and if the guest was rebooted 120 secs back, this metric will report a value of 120 seconds. The accuracy of this metric is dependent on the measurement period – the smaller the measurement period, greater the accuracy.
Total uptime of the VM	Indicates the total time that the VM has been up since its last reboot.	Mins	Administrators may wish to be alerted if a guest has been running without a reboot for a very long period. Setting a threshold for this metric allows administrators to determine such conditions.

3.7.5 Memory Usage - VM Test

This test reports statistics related to the usage of physical memory of the VMs.

Target of the test:

A Hyper-V / Hyper-V VDI server

Agent executing the test:

An internal agent

Output of the test:

For a Hyper-V server, one set of results will be reported for every VM on the server

For a Hyper-V VDI server, one set of results will be reported for the user who is currently logged into each virtual desktop on the server

Parameters of the test:

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured.
3. **PORT** - The port at which the **HOST** listens. By default, this is NULL.
4. **INSIDE VIEW USING** - By default, this test communicates with every VM remotely and extracts "inside view" metrics. Therefore, by default, the **INSIDE VIEW USING** flag is set to **Remote connection to VM (Windows)**.

   Typically, to establish this remote connection, eG Enterprise requires that the eG agent be configured with domain administrator privileges. In high-security environments, where the IT staff might have reservations about exposing the credentials of their domain administrators, this approach to extracting "inside view" metrics might not be preferred. In such environments therefore, eG Enterprise provides administrators the option to deploy a piece of software called the **eG VM Agent** on every Windows VM; this VM agent allows the eG agent to collect "inside view" metrics from the Windows VMs **without domain administrator rights.** Refer to Section 1.3.2 for more details on the **eG VM Agent**. To ensure that the "inside view" of Windows VMs is obtained using the eG VM Agent, set the **INSIDE VIEW USING** flag to **eG VM Agent (Windows)**. Once this is done, you can set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.
5. **DOMAIN, ADMIN USER, ADMIN PASSWORD,** and **CONFIRM PASSWORD** – By default, this test connects to each virtual guest remotely and attempts to collect "inside view" metrics. Accordingly, the **INSIDE VIEW USING** flag is set to **Remote connection to VM (Windows)** by default. To obtain a remote connection, the test must be configured with the privileges of an administrative user to the domain within which the guests reside. The first step towards this is to specify the DOMAIN within which the virtual guests reside. The **ADMIN USER** and **ADMIN PASSWORD** will change according to the **DOMAIN** specification. Discussed below are the different values that the **DOMAIN** parameter can take, and how they impact the **ADMIN USER** and **ADMIN PASSWORD** specifications:
   - **If the VMs belong to a single domain:** If the guests belong to a specific domain, then specify the name of that domain against the domain parameter. In this case, any administrative user in that domain will have remote access to all the virtual guests. Therefore, an administrator account in the given domain can be provided in the **ADMIN USER** field and the corresponding password in the **ADMIN PASSWORD** field. Confirm the password by retyping it in the **CONFIRM PASSWORD** text box.
   - I**f the VMs belong to different domains:**In this case, you might want to provide multiple domain names. If this is done, then, to access the guests in every configured domain, the test should be configured with the required user privileges; this implies that along with multiple **DOMAIN** names, multiple **ADMIN USER** names and **ADMIN PASSWORDS** would also have to be provided. To help administrators provide these user details quickly and easily, the eG administrative interface embeds a special configuration page. To access this page, simply click on the encircled '+' icon that appears alongside the . To know how to use the special page, refer to Section 3.6.1 of this document.

- **If the INSIDE VIEW USING flag is set to 'eG VM Agent (Windows)'**: On the other hand, if the **INSIDE VIEW USING** flag is set to **eG VM Agent (Windows)**, then it implies that the inside view can be obtained without domain administrator privileges. Therefore, set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.

4. **REPORT BY USER** – For the *Hyper-V* monitoring model, the REPORT BY USER flag is set to **NO** by default, indicating that by default, the guest operating systems on the Hyper-V server are identified using the hostname specified in the operating system. On the other hand, for the *Hyper-V VDI* model, this flag is set to **YES** by default; this implies that in case of VDI servers, by default, the guests will be identified using the login of the user who is accessing the guest OS. In other words, in VDI environments, this test will, by default, report measures for every *username_on_virtualmachinename*.
5. **REPORT POWERED OS** - **This flag becomes relevant only if the REPORT BY USER flag is set to 'Yes'.**

   If the **REPORT POWERED OS** flag is set to **Yes** (which is the default setting), then this test will report measures for even those VMs that do not have any users logged in currently. Such guests will be identified by their *virtualmachine name* and not by the *username_on_virtualmachinename*. On the other hand, if the **REPORT POWERED OS** flag is set to **No**, then this test will not report measures for those VMs to which no users are logged in currently.
6. **IGNORE VMS INSIDE VIEW** - Administrators of some high security Hyper-V environments might not have permissions to internally monitor one/more VMs. The eG agent can be configured to **not obtain the 'inside view' of such 'inaccessible' VMs** using the **IGNORE VMS INSIDE VIEW** parameter. Against this parameter, you can provide a comma-separated list of VM names, or VM name patterns, for which the inside view need not be obtained. For instance, your **IGNORE VMS INSIDE VIEW** specification can be: **xp,*lin*,win*,vista*. Here, the * (asterisk) is used to denote leading and trailing spaces (as the case may be). By default, this parameter is set to *none* indicating that the eG agent obtains the inside view of all VMs on a Hyper-V host by default.

   Note:
   While performing VM discovery, the eG agent will not discover the operating system of the VMs configured in the **IGNORE VMS INSIDE VIEW** text box.
7. **EXCLUDE VMS** - Administrators of some virtualized environments may not want to monitor some of their less-critical VMs - for instance, VM templates - both from 'outside' and from 'inside'. The eG agent in this case can be configured to completely exclude such VMs from its monitoring purview. To achieve this, provide a comma-separated list of VMs to be excluded from monitoring in the **EXCLUDE VMS** text box. Instead of VMs, VM name patterns can also be provided here in a comma-separated list. For example, your **EXCLUDE VMS** specification can be: **xp,*lin*,win*,vista*. Here, the * (asterisk) is used to denote leading and trailing spaces (as the case may be). By default, this parameter is set to *none* indicating that the eG agent obtains the inside and outside views of all VMs on a virtual host by default. By providing a comma-separated list of VMs/VM name patterns in the **EXCLUDE VMS** text box, you can make sure the eG agent stops collecting 'inside' and 'outside' view metrics for a configured set of VMs.
8. **IGNORE WINNT** – By default, the eG agent does not support the *inside view* for VMs executing on **Windows NT** operating systems. Accordingly, the **IGNORE WINNT** flag is set to **Yes** by default.

9. **DETAILED DIAGNOSIS** - To make diagnosis more efficient and accurate, the eG suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option.

   The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:

   - The eG manager license should allow the detailed diagnosis capability
   - Both the normal and abnormalfrequencies configured for the detailed diagnosis measures should not be 0.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
Total physical memory	Indicates the total physical memory of this VM.	MB	
Used physical memory	Indicates the used physical memory of this VM.	MB	
Free physical memory	Indicates the free physical memory of the VM.	MB	This measure typically indicates the amount of memory available for use by applications running on the target VM. On Unix operating systems (AIX and Linux), the operating system tends to use parts of the available memory for caching files, objects, etc. When applications require additional memory, this is released from the operating system cache. Hence, to understand the true free memory that is available to applications, the eG agent reports the sum of the free physical memory and the operating system cache memory size as the value of the Free physical memory measure while monitoring AIX and Linux guest operating systems.

Measurement	Description	Measurement Unit	Interpretation
Physical memory utilized	Indicates the percent usage of physical memory by this VM.	Percent	Ideally, the value of this measure should be low. While sporadic spikes in memory usage could be caused by one/more rogue processes on the VM, a consistent increase in this value could be a cause for some serious concern, as it indicates a gradual, but steady erosion of valuable memory resources. If this unhealthy trend is not repaired soon, it could severely hamper VM performance, causing anything from a slowdown to a complete system meltdown. You can use the detailed diagnosis of this measure to figure out which processes on the VM are consuming memory excessively.
Available physical memory	Indicates the amount of physical memory, immediately available for allocation to a process or for system use.	MB	Not all of the Available physical memoryisFree physical memory. Typically,Available physical memoryismade up of theStandby List, Free List, andZeroed List. When Windows wants to trim a process' working set, the trimmed pages are moved (usually) to the Standby List. From here, they can be brought back to life in the working set with only a soft page fault (much faster than a hard fault, which would have to talk to the disk). If a page stays in the standby List for a long time, it gets freed and moved to the Free List. In the background, there is a low priority thread (actually, the only thread with priority 0) which takes pages from the Free List and zeros them out. Because of this, there is usually very little in the Free List.

Measurement	Description	Measurement Unit	Interpretation
			All new allocations always come from the Zeroed List, which is memory pages that have been overwritten with zeros. This is a standard part of the OS' cross-process security, to prevent any process ever seeing data from another. If the Zeroed List is empty, Free List memory is zeroed and used or, if that is empty too, Standby List memory is freed, zeroed, and used. It is because all three can be used with so little effort that they are all counted as "available". A high value is typically desired for this measure. This measure will be available for Windows 2008 VMs only.
Modified memory	Indicates the amount of memory that is allocated to the modified page list.	MB	This memory contains cached data and code that is not actively in use by processes, the system and the system cache. This memory needs to be written out before it will be available for allocation to a process or for system use. Cache pages on the modified list have been altered in memory. No process has specifically asked for this data to be in memory, it is merely there as a consequence of caching. Therefore it can be written to disk at any time (not to the page file, but to its original file location) and reused. However, since this involves I/O, it is not considered to be Available physical memory. This measure will be available for Windows 2008 VMs only.

Measurement	Description	Measurement Unit	Interpretation
Standby memory:	Indicates the amount of memory assigned to the standby list.	MB	This memory contains cached data and code that is not actively in use by processes, the system and the system cache. It is immediately available for allocation to a process or for system use. If the system runs out of available free and zero memory, memory on lower priority standby cache page lists will be repurposed before memory on higher priority standby cache page lists. Typically, Standby memory is the aggregate of Standby Cache Core Bytes,Standby Cache Normal Priority Bytes, and Standby Cache Reserve Bytes. Standby Cache Core Bytes is the amount of physical memory, that is assigned to the core standby cache page lists. Standby Cache Normal Priority Bytes is the amount of physical memory, that is assigned to the normal priority standby cache page lists. Standby Cache Reserve Bytes is the amount of physical memory, that is assigned to the reserve standby cache page lists. This measure will be available for Windows 2008 VMs only.
Cached memory:	This measure is an aggregate of Standby memory and Modified memory.	MB	This measure will be available for Windows 2008 VMs only.

Note:

While monitoring Linux/AIX guest operating systems, you may observe discrepancies between the value of the *Physical memory utilized* measure and the memory usage percentages reported per process by the detailed diagnosis of the same measure. This is because, while the *Physical memory utilized*measure takes into account the memory in the OS cache of the Linux/AIX VM, the memory usage percent that the detailed diagnosis reports per process does not consider the OS cache memory.

3.7.6 Windows Memory - VM Test

To understand the metrics reported by this test, it is essential to understand how memory is handled by the operating system. On any Windows system, memory is partitioned into a part that is available for user processes, and another that is available to the OS kernel. The kernel memory area is divided into several parts, with the two major parts (called "pools") being a nonpaged pool and a paged pool. The nonpaged pool is a section of memory that cannot, under any circumstances, be paged to disk. The paged pool is a section of memory that can be paged to disk. (Just being stored in the paged pool doesn't necessarily mean that something has been paged to disk. It just means that it has either been paged to disk or it could be paged to disk.) Sandwiched directly in between the nonpaged and paged pools (although technically part of the nonpaged pool) is a section of memory called the "System Page Table Entries," or "System PTEs." The WindowsMemory – Guest test tracks critical metrics corresponding to the System PTEs and the pool areas of kernel memory of a Windows virtual machine.

Target of the test	**A Hyper-V host**
Agent deploying the test	**An internal agent**
Configurable parameters for the test	1. **TESTPERIOD** - How often should the test be executed **HOST** - The host for which the test is to be configured. **PORT** - The port at which the **HOST** listens. By default, this is NULL. **INSIDE VIEW USING** - By default, this test communicates with every VM remotely and extracts “inside view” metrics. Therefore, by default, the **INSIDE VIEW USING** flag is set to **Remote connection to VM (Windows)**. Typically, to establish this remote connection, eG Enterprise requires that the eG agent be configured with domain administrator privileges. In high-security environments, where the IT staff might have reservations about exposing the credentials of their domain administrators, this approach to extracting “inside view” metrics might not be preferred. In such environments therefore, eG Enterprise provides administrators the option to deploy a piece of software called the **eG VM Agent** on every Windows VM; this VM agent allows the eG agent to collect “inside view” metrics from the Windows VMs **without domain administrator rights.** Refer to Section 1.3.2 for more details on the **eG VM Agent**. To ensure that the “inside view” of Windows VMs is obtained using the eG VM Agent, set the **INSIDE VIEW USING** flag to **eG VM Agent (Windows)**. Once this is done, you can set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*. **DOMAIN, ADMIN USER, ADMIN PASSWORD,** and **CONFIRM PASSWORD** – By default, this test connects to each virtual guest remotely and attempts to collect “inside view” metrics. Accordingly, the **INSIDE VIEW USING** flag is set to **Remote connection to VM (Windows)** by default. To obtain a remote connection, the test must be configured with the privileges of an administrative user to the domain within which the guests reside. The first step towards this is to specify the DOMAIN within which the virtual guests reside. The **ADMIN USER** and **ADMIN PASSWORD** will change according to the **DOMAIN** specification. Discussed

	below are the different values that the **DOMAIN** parameter can take, and how they impact the **ADMIN USER** and **ADMIN PASSWORD** specifications: • **If the VMs belong to a single domain**: **If the guests belong to a specific domain, then specify the name of that domain against the DOMAIN** parameter. In this case, any **administrative user in that domain will have remote access to all the virtual guests. Therefore, an administrator account in the given domain can be provided in the** ADMIN USER **field and the corresponding password in the** ADMIN PASSWORD **field. Confirm the password by retyping it in the** CONFIR**M PASSWORD text box.** • If the VMs belong to different domains: In this case, you might want to provide multiple domain names. If this is done, then, to access the guests in every configured domain, the test should be configured with the required user privileges; this implies that along with multiple **DOMAIN** names, multiple **ADMIN USER** names and **ADMIN PASSWORDS** would also have to be provided. To help administrators provide these user details quickly and easily, the eG administrative interface embeds a special configuration page. To access this page, simply click on the **Click here** hyperlink that appears just above the parameters of this test in the test configuration page. To know how to use the special page, refer to Section 3.6.1 of this document.

	• **If the INSIDE VIEW USING flag is set to 'eG VM Agent (Windows)'**: On the other hand, if the **INSIDE VIEW USING** flag is set to **eG VM Agent (Windows)**, then it implies that the inside view can be obtained without domain administrator privileges. Therefore, set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*. **REPORT BY USER** – For the *Hyper-V* monitoring model, the REPORT BY USER flag is set to **NO** by default, indicating that by default, the guest operating systems on the Hyper-V server are identified using the hostname specified in the operating system. On the other hand, for the *Hyper-V VDI* model, this flag is set to **YES** by default; this implies that in case of VDI servers, by default, the guests will be identified using the login of the user who is accessing the guest OS. In other words, in VDI environments, this test will, by default, report measures for every *username_on_virtualmachinename*. **REPORT POWERED OS** - **This flag becomes relevant only if the REPORT BY USER flag is set to 'Yes'.**

	If the **REPORT POWERED OS** flag is set to **Yes** (which is the default setting), then this test will report measures for even those VMs that do not have any users logged in currently. Such guests will be identified by their *virtualmachine name* and not by the *username_on_virtualmachinename*. On the other hand, if the **REPORT POWERED OS** flag is set to **No**, then this test will not report measures for those VMs to which no users are logged in currently. **IGNORE VMS INSIDE VIEW** - Administrators of some high security Hyper-V environments might not have permissions to internally monitor one/more VMs. The eG agent can be configured to **not obtain the 'inside view' of such 'inaccessible' VMs** using the **IGNORE VMS INSIDE VIEW** parameter. Against this parameter, you can provide a comma-separated list of VM names, or VM name patterns, for which the inside view need not be obtained. For instance, your **IGNORE VMS INSIDE VIEW** specification can be: **xp,*lin*,win*,vista*. Here, the * (asterisk) is used to denote leading and trailing spaces (as the case may be). By default, this parameter is set to *none* indicating that the eG agent obtains the inside view of all VMs on a Hyper-V host by default. **Note:** While performing VM discovery, the eG agent will not discover the operating system of the VMs configured in the **IGNORE VMS INSIDE VIEW** text box.
	EXCLUDE VMS - Administrators of some virtualized environments may not want to monitor some of their less-critical VMs - for instance, VM templates - both from 'outside' and from 'inside'. The eG agent in this case can be configured to completely exclude such VMs from its monitoring purview. To achieve this, provide a comma-separated list of VMs to be excluded from monitoring in the **EXCLUDE VMS** text box. Instead of VMs, VM name patterns can also be provided here in a comma-separated list. For example, your **EXCLUDE VMS** specification can be: **xp,*lin*,win*,vista*. Here, the * (asterisk) is used to denote leading and trailing spaces (as the case may be). By default, this parameter is set to *none* indicating that the eG agent obtains the inside and outside views of all VMs on a virtual host by default. By providing a comma-separated list of VMs/VM name patterns in the **EXCLUDE VMS** text box, you can make sure the eG agent stops collecting 'inside' and 'outside' view metrics for a configured set of VMs. **IGNORE WINNT** – By default, the eG agent does not support the *inside view* for VMs executing on **Windows NT** operating systems. Accordingly, the **IGNORE WINNT** flag is set to **Yes** by default.
Outputs of the	**One set of results for every Windows VM guest/user on the monitored Hyper-V**

test	**server**

Measurements made by the test	Measurement	Measurement Unit	Interpretation
	Free entries in system page table: **Indicates the *number of page table entries not currently in use by the guest.***	**Number**	The maximum number of System PTEs that a server can have is set when the server boots. In heavily-used servers, you can run out of system PTEs. You can use the registry to increase the number of system PTEs, but that encroaches into the paged pool area, and you could run out of paged pool memory. Running out of either one is bad, and the goal should be to tune your server so that you run out of both at the exact same time. Typically, the value of this metric should be above 3000.
	Page read rate in VM: Indicates the average number of times per second the disk was read to resolve hard fault paging.	**Reads/Sec**	
	Page write rate in VM: Indicates the average number of times per second the pages are written to disk to free up the physical memory.	**Writes/Sec**	
	Page input rate in VM: Indicates the number of times per second that a process needed to access a piece of memory that was not in its working set, meaning that the guest had to retrieve it from the page file.	**Pages/Sec**	
	Page output rate in VM: Indicates the number of times per second the guest decided to trim a process's working set by writing some memory to disk in order to free up physical memory for another process.	**Pages/Sec**	This value is a critical measure of the memory utilization on a guest. If this value never increases, then there is sufficient memory in the guest. Instantaneous spikes of this value are acceptable, but if the value itself starts to rise over time or with load, it implies that there is a memory shortage on the guest.

	Memory pool non-paged data in VM: Indicates the total size of the kernel memory nonpaged pool.	**MB**	The kernel memory nonpage pool is an area of guest memory (that is, memory used by the guest operating system) for kernel objects that cannot be written to disk, but must remain in memory as long as the objects are allocated. Typically, there should be no more than 100 MB of non-paged pool memory being used.
	Memory pool paged data in VM : Indicates the total size of the Paged Pool.	**MB**	If the Paged Pool starts to run out of space (when it's 80% full by default), the guest will automatically take some memory away from the System File Cache and give it to the Paged Pool. This makes the System File Cache smaller. However, the system file cache is critical, and so it will never reach zero. Hence, a significant increase in the paged pool size is a problem. This metric is a useful indicator of memory leaks in a guest. A memory leak occurs when the guest allocates more memory to a process than the process gives back to the pool. Any time of process can cause a memory leak. If the amount of paged pool data keeps increasing even though the workload on the guest remains constant, it is an indicator of a memory leak.

3.7.7 Windows Network Traffic - VM Test

This is an internal test that monitors the incoming and outgoing traffic through each Windows guest of a Hyper-V server.

Target of the test:

A Hyper-V / Hyper-V VDI server

Agent executing the test:

An internal agent

Output of the test:

For a Hyper-V server, one set of results will be reported for every network interface supported by each Windows VM on the server.

For a Hyper-V VDI server, one set of results will be reported for every network interface used by the user who is logged into each Windows virtual desktop on the server

First-level descriptor: Windows VM or User on Windows virtual desktop

Second-level descriptor: Network interface

Parameters of the test:

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured.
3. **PORT** - The port at which the **HOST** listens. By default, this is NULL.
4. **INSIDE VIEW USING** - By default, this test communicates with every VM remotely and extracts "inside view" metrics. Therefore, by default, the **INSIDE VIEW USING** flag is set to **Remote connection to VM (Windows)**.

   Typically, to establish this remote connection, eG Enterprise requires that the eG agent be configured with domain administrator privileges. In high-security environments, where the IT staff might have reservations about exposing the credentials of their domain administrators, this approach to extracting "inside view" metrics might not be preferred. In such environments therefore, eG Enterprise provides administrators the option to deploy a piece of software called the **eG VM Agent** on every Windows VM; this VM agent allows the eG agent to collect "inside view" metrics from the Windows VMs **without domain administrator rights.** Refer to Section 1.3.2 for more details on the **eG VM Agent**. To ensure that the "inside view" of Windows VMs is obtained using the eG VM Agent, set the **INSIDE VIEW USING** flag to **eG VM Agent (Windows)**. Once this is done, you can set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.
5. **DOMAIN, ADMIN USER, ADMIN PASSWORD,** and **CONFIRM PASSWORD** – By default, this test connects to each virtual guest remotely and attempts to collect "inside view" metrics. Accordingly, the **INSIDE VIEW USING** flag is set to **Remote connection to VM (Windows)** by default. To obtain a remote connection, the test must be configured with the privileges of an administrative user to the domain within which the guests reside. The first step towards this is to specify the DOMAIN within which the virtual guests reside. The **ADMIN USER** and **ADMIN PASSWORD** will change according to the **DOMAIN** specification. Discussed below are the different values that the **DOMAIN** parameter can take, and how they impact the **ADMIN USER** and **ADMIN PASSWORD** specifications:
   - **If the VMs belong to a single domain:** If the guests belong to a specific domain, then specify the name of that domain against the domain parameter. In this case, any administrative user in that domain will have remote access to all the virtual guests. Therefore, an administrator account in the given domain can be provided in the **ADMIN USER** field and the corresponding password in the **ADMIN PASSWORD** field. Confirm the password by retyping it in the **CONFIRM PASSWORD** text box.

- **If the VMs belong to different domains:**In this case, you might want to provide multiple domain names. If this is done, then, to access the guests in every configured domain, the test should be configured with the required user privileges; this implies that along with multiple **DOMAIN** names, multiple **ADMIN USER** names and **ADMIN PASSWORDS** would also have to be provided. To help administrators provide these user details quickly and easily, the eG administrative interface embeds a special configuration page. To access this page, simply click on the encircled '+' icon that appears alongside the . To know how to use the special page, refer to Section 3.6.1 of this document.
- **If the INSIDE VIEW USING flag is set to 'eG VM Agent (Windows)'**: On the other hand, if the **INSIDE VIEW USING** flag is set to **eG VM Agent (Windows)**, then it implies that the inside view can be obtained without domain administrator privileges. Therefore, set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.

4. **REPORT BY USER** – For the *Hyper-V* monitoring model, the REPORT BY USER flag is set to **NO** by default, indicating that by default, the guest operating systems on the Hyper-V server are identified using the hostname specified in the operating system. On the other hand, for the *Hyper-V VDI* model, this flag is set to **YES** by default; this implies that in case of VDI servers, by default, the guests will be identified using the login of the user who is accessing the guest OS. In other words, in VDI environments, this test will, by default, report measures for every *username_on_virtualmachinename*.
5. **REPORT POWERED OS** - **This flag becomes relevant only if the REPORT BY USER flag is set to 'Yes'.**

   If the **REPORT POWERED OS** flag is set to **Yes** (which is the default setting), then this test will report measures for even those VMs that do not have any users logged in currently. Such guests will be identified by their *virtualmachine name* and not by the *username_on_virtualmachinename*. On the other hand, if the **REPORT POWERED OS** flag is set to **No**, then this test will not report measures for those VMs to which no users are logged in currently.
6. **IGNORE VMS INSIDE VIEW** - Administrators of some high security Hyper-V environments might not have permissions to internally monitor one/more VMs. The eG agent can be configured to **not obtain the 'inside view' of such 'inaccessible' VMs** using the **IGNORE VMS INSIDE VIEW** parameter. Against this parameter, you can provide a comma-separated list of VM names, or VM name patterns, for which the inside view need not be obtained. For instance, your **IGNORE VMS INSIDE VIEW** specification can be: **xp,*lin*,win*,vista*. Here, the * (asterisk) is used to denote leading and trailing spaces (as the case may be). By default, this parameter is set to *none* indicating that the eG agent obtains the inside view of all VMs on a Hyper-V host by default.

   Note:
   While performing VM discovery, the eG agent will not discover the operating system of the VMs configured in the **IGNORE VMS INSIDE VIEW** text box.
7. **EXCLUDE VMS** - Administrators of some virtualized environments may not want to monitor some of their less-critical VMs - for instance, VM templates - both from 'outside' and from 'inside'. The eG agent in this case can be configured to completely exclude such VMs from its monitoring purview. To achieve this, provide a comma-separated list of VMs to be excluded from monitoring in the **EXCLUDE VMS** text box. Instead of VMs, VM name patterns can also be provided here in a comma-separated list. For

example, your **EXCLUDE VMS** specification can be: **xp,*lin*,win*,vista*. Here, the * (asterisk) is used to denote leading and trailing spaces (as the case may be). By default, this parameter is set to *none* indicating that the eG agent obtains the inside and outside views of all VMs on a virtual host by default. By providing a comma-separated list of VMs/VM name patterns in the **EXCLUDE VMS** text box, you can make sure the eG agent stops collecting 'inside' and 'outside' view metrics for a configured set of VMs.

8. **IGNORE WINNT** – By default, the eG agent does not support the *inside view* for VMs executing on **Windows NT** operating systems. Accordingly, the **IGNORE WINNT** flag is set to **Yes** by default.
9. **IS FULL DUPLEX** - By default, this flag is set to **Yes**, indicating that the incoming and outgoing data traffic is handled in full duplex mode. If the data traffic in your environment is handled in half-duplex mode, set this flag to **No**.
10. **DETAILED DIAGNOSIS** - To make diagnosis more efficient and accurate, the eG suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option.

    The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:

    - The eG manager license should allow the detailed diagnosis capability
    - Both the normal and abnormalfrequencies configured for the detailed diagnosis measures should not be 0.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
Incoming traffic	Indicates the rate at which data (including framing characters) is received on a network interface.	Mbps	An abnormally high rate of incoming traffic may require additional analysis.
Outgoing traffic	Represents the rate at which data (including framing characters) is sent on a network interface.	Mbps	An abnormally high rate of outgoing traffic may require additional analysis.
Maximum bandwidth	An estimate of the capacity of a network interface.	Mbps	

Measurement	Description	Measurement Unit	Interpretation
Bandwidth usage	Indicates the percentage of bandwidth used by a network interface.	Percent	By comparing the bandwidth usage with the maximum bandwidth of an interface, an administrator can determine times when the network interface is overloaded or is being a performance bottleneck.
Output queue length	Indicates the length of the output packet queue (in packets)	Number	If this is longer than 2, delays are being experienced and the bottleneck should be found and eliminated if possible.
Outbound packet errors	The number of outbound packets that could not be transmitted because of errors	Number	Ideally, number of outbound errors should be 0.
Inbound packet errors	The number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol.	Number	Ideally, number of inbound errors should be 0.

If the WindowsNetTraffic – Guest test is not reporting measures for a guest, make sure that you have enabled the SNMP service for the guest.

3.7.8 TCP - VM Test

This test tracks various statistics pertaining to TCP connections to and from each guest on a Hyper-V host. The details of the test are provided below:

Target of the test:

A Hyper-V / Hyper-V VDI server

Agent executing the test:

An internal agent

Output of the test:

For a Hyper-V server, one set of results will be reported for every powered-on VM on the server.

For a Hyper-V VDI server, one set of results will be reported for the user who is currently logged into each virtual desktop on the server

Parameters of the test:

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured.
3. **PORT** - The port at which the **HOST** listens. By default, this is NULL.
4. **INSIDE VIEW USING** - By default, this test communicates with every VM remotely and extracts "inside view" metrics. Therefore, by default, the **INSIDE VIEW USING** flag is set to **Remote connection to VM (Windows)**.

   Typically, to establish this remote connection, eG Enterprise requires that the eG agent be configured with domain administrator privileges. In high-security environments, where the IT staff might have reservations about exposing the credentials of their domain administrators, this approach to extracting "inside view" metrics might not be preferred. In such environments therefore, eG Enterprise provides administrators the option to deploy a piece of software called the **eG VM Agent** on every Windows VM; this VM agent allows the eG agent to collect "inside view" metrics from the Windows VMs **without domain administrator rights.** Refer to Section 1.3.2 for more details on the **eG VM Agent**. To ensure that the "inside view" of Windows VMs is obtained using the eG VM Agent, set the **INSIDE VIEW USING** flag to **eG VM Agent (Windows)**. Once this is done, you can set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.
5. **DOMAIN, ADMIN USER, ADMIN PASSWORD,** and **CONFIRM PASSWORD** – By default, this test connects to each virtual guest remotely and attempts to collect "inside view" metrics. Accordingly, the **INSIDE VIEW USING** flag is set to **Remote connection to VM (Windows)** by default. To obtain a remote connection, the test must be configured with the privileges of an administrative user to the domain within which the guests reside. The first step towards this is to specify the DOMAIN within which the virtual guests reside. The **ADMIN USER** and **ADMIN PASSWORD** will change according to the **DOMAIN** specification. Discussed below are the different values that the **DOMAIN** parameter can take, and how they impact the **ADMIN USER** and **ADMIN PASSWORD** specifications:
   - **If the VMs belong to a single domain:** If the guests belong to a specific domain, then specify the name of that domain against the domain parameter. In this case, any administrative user in that domain will have remote access to all the virtual guests. Therefore, an administrator account in the given domain can be provided in the **ADMIN USER** field and the corresponding password in the **ADMIN PASSWORD** field. Confirm the password by retyping it in the **CONFIRM PASSWORD** text box.
   - I**f the VMs belong to different domains:**In this case, you might want to provide multiple domain names. If this is done, then, to access the guests in every configured domain, the test should be configured with the required user privileges; this implies that along with multiple **DOMAIN** names, multiple **ADMIN USER** names and **ADMIN PASSWORDS** would also have to be provided. To help administrators provide these user details quickly and easily, the eG administrative interface embeds a

special configuration page. To access this page, simply click on the encircled '+' icon that appears alongside the . To know how to use the special page, refer to Section 3.6.1 of this document.

- **If the INSIDE VIEW USING flag is set to 'eG VM Agent (Windows)'**: On the other hand, if the **INSIDE VIEW USING** flag is set to **eG VM Agent (Windows)**, then it implies that the inside view can be obtained without domain administrator privileges. Therefore, set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.

4. **REPORT BY USER** – For the *Hyper-V* monitoring model, the REPORT BY USER flag is set to **NO** by default, indicating that by default, the guest operating systems on the Hyper-V server are identified using the hostname specified in the operating system. On the other hand, for the *Hyper-V VDI* model, this flag is set to **YES** by default; this implies that in case of VDI servers, by default, the guests will be identified using the login of the user who is accessing the guest OS. In other words, in VDI environments, this test will, by default, report measures for every *username_on_virtualmachinename*.

5. **REPORT POWERED OS** - **This flag becomes relevant only if the REPORT BY USER flag is set to 'Yes'.**

   If the **REPORT POWERED OS** flag is set to **Yes** (which is the default setting), then this test will report measures for even those VMs that do not have any users logged in currently. Such guests will be identified by their *virtualmachine name* and not by the *username_on_virtualmachinename*. On the other hand, if the **REPORT POWERED OS** flag is set to **No**, then this test will not report measures for those VMs to which no users are logged in currently.

6. **IGNORE VMS INSIDE VIEW** - Administrators of some high security Hyper-V environments might not have permissions to internally monitor one/more VMs. The eG agent can be configured to **not obtain the 'inside view' of such 'inaccessible' VMs** using the **IGNORE VMS INSIDE VIEW** parameter. Against this parameter, you can provide a comma-separated list of VM names, or VM name patterns, for which the inside view need not be obtained. For instance, your **IGNORE VMS INSIDE VIEW** specification can be: **xp,*lin*,win*,vista*. Here, the * (asterisk) is used to denote leading and trailing spaces (as the case may be). By default, this parameter is set to *none* indicating that the eG agent obtains the inside view of all VMs on a Hyper-V host by default.

   Note:
   While performing VM discovery, the eG agent will not discover the operating system of the VMs configured in the **IGNORE VMS INSIDE VIEW** text box.

7. **EXCLUDE VMS** - Administrators of some virtualized environments may not want to monitor some of their less-critical VMs - for instance, VM templates - both from 'outside' and from 'inside'. The eG agent in this case can be configured to completely exclude such VMs from its monitoring purview. To achieve this, provide a comma-separated list of VMs to be excluded from monitoring in the **EXCLUDE VMS** text box. Instead of VMs, VM name patterns can also be provided here in a comma-separated list. For example, your **EXCLUDE VMS** specification can be: **xp,*lin*,win*,vista*. Here, the * (asterisk) is used to denote leading and trailing spaces (as the case may be). By default, this parameter is set to *none* indicating that the eG agent obtains the inside and outside views of all VMs on a virtual host by default. By providing a comma-separated list of VMs/VM name patterns in the **EXCLUDE VMS** text box, you can make sure the eG agent stops collecting 'inside' and 'outside' view metrics for a configured set of VMs.

8. **IGNORE WINNT** – By default, the eG agent does not support the *inside view* for VMs executing on **Windows NT** operating systems. Accordingly, the **IGNORE WINNT** flag is set to **Yes** by default.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
Incoming connections to VM	Indicates the connections per second received by the guest.	Conns/Sec	A high value can indicate an increase in input load.
Outgoing connections to VM	Indicates the connections per second initiated by the guest.	Conns/Sec	A high value can indicate that one or more of the applications executing on the guest have started using a number of TCP connections to some other guest or host.
Current connections to VM	Indicates the currently established connections.	Number	A sudden increase in the number of connections established on a guest can indicate either an increase in

Measurement	Description	Measurement Unit	Interpretation
			load to one or more of the applications executing on the guest, or that one or more of the applications are experiencing a problem (e.g., a slow down). On Microsoft Windows, the current connections metrics is the total number of TCP connections that are currently in the ESTABLISHED or CLOSE_WAIT states.
Connection drops on VM	Indicates the rate of established TCP connections dropped from the TCP listen queue.	Conns/Sec	This value should be 0 for most of the time. Any non-zero value implies that one or more applications on the guest are under overload.
Connection failures on VM	Indicates the rate of half open TCP connections dropped from the listen queue.	Conns/Sec	This value should be 0 for most of the time. A prolonged non-zero value can indicate either that the server is under SYN attack or that there is a problem with the network link to the server that is resulting in connections being dropped without completion.

3.7.9 TCP Traffic - VM Test

Since most popular applications rely on the TCP protocol for their proper functioning, traffic monitoring at the TCP protocol layer can provide good indicators of the performance seen by the applications that use TCP. The most critical metric at the TCP protocol layer is the percentage of retransmissions. Since TCP uses an exponential back-off algorithm for its retransmissions, any retransmission of packets over the network (due to network congestion, noise, data link errors, etc.) can have a significant impact on the throughput seen by applications that use TCP. This test monitors the TCP protocol traffic to and from a guest, and particularly monitors retransmissions.

Target of the test:

A Hyper-V / Hyper-V VDI server

Agent executing the test:

An internal agent

Output of the test:

For a Hyper-V server, one set of results will be reported for every powered-on VM on the server.

For a Hyper-V VDI server, one set of results will be reported for the user who is currently logged into each virtual desktop on the server

Parameters of the test:

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured.
3. **PORT** - The port at which the **HOST** listens. By default, this is NULL.
4. **INSIDE VIEW USING** - By default, this test communicates with every VM remotely and extracts "inside view" metrics. Therefore, by default, the **INSIDE VIEW USING** flag is set to **Remote connection to VM (Windows)**.

   Typically, to establish this remote connection, eG Enterprise requires that the eG agent be configured with domain administrator privileges. In high-security environments, where the IT staff might have reservations about exposing the credentials of their domain administrators, this approach to extracting "inside view" metrics might not be preferred. In such environments therefore, eG Enterprise provides administrators the option to deploy a piece of software called the **eG VM Agent** on every Windows VM; this VM agent allows the eG agent to collect "inside view" metrics from the Windows VMs **without domain administrator rights.** Refer to Section 1.3.2 for more details on the **eG VM Agent**. To ensure that the "inside view" of Windows VMs is obtained using the eG VM Agent, set the **INSIDE VIEW USING** flag to **eG VM Agent (Windows)**. Once this is done, you can set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.
5. **DOMAIN, ADMIN USER, ADMIN PASSWORD,** and **CONFIRM PASSWORD** – By default, this test connects to each virtual guest remotely and attempts to collect "inside view" metrics. Accordingly, the **INSIDE VIEW USING** flag is set to **Remote connection to VM (Windows)** by default. To obtain a remote connection, the test must be configured with the privileges of an administrative user to the domain within which the guests reside. The first step towards this is to specify the DOMAIN within which the virtual guests reside. The **ADMIN USER** and **ADMIN PASSWORD** will change according to the **DOMAIN** specification. Discussed below are the different values that the **DOMAIN** parameter can take, and how they impact the **ADMIN USER** and **ADMIN PASSWORD** specifications:
   - **If the VMs belong to a single domain:** If the guests belong to a specific domain, then specify the name of that domain against the domain parameter. In this case, any administrative user in that domain will have remote access to all the virtual guests. Therefore, an administrator account in the given domain can be provided in the **ADMIN USER** field and the corresponding password in the **ADMIN**

PASSWORD field. Confirm the password by retyping it in the **CONFIRM PASSWORD** text box.

- I**f the VMs belong to different domains:**In this case, you might want to provide multiple domain names. If this is done, then, to access the guests in every configured domain, the test should be configured with the required user privileges; this implies that along with multiple **DOMAIN** names, multiple **ADMIN USER** names and **ADMIN PASSWORDS** would also have to be provided. To help administrators provide these user details quickly and easily, the eG administrative interface embeds a special configuration page. To access this page, simply click on the encircled '+' icon that appears alongside the . To know how to use the special page, refer to Section 3.6.1 of this document.
- **If the INSIDE VIEW USING flag is set to 'eG VM Agent (Windows)'**: On the other hand, if the **INSIDE VIEW USING** flag is set to **eG VM Agent (Windows)**, then it implies that the inside view can be obtained without domain administrator privileges. Therefore, set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.

4. **SEGMENTS SENT MIN** - Specify the minimum threshold for the number of segments sent/transmitted over the network. The default value is 10; in this case, the test will compute/report the *Retransmit ratio from VM* measure only if more than 10 segments are sent over the network – i.e., if the value of the *Segments sent by VM* measure crosses the value 10. On the other hand, if the *Segments sent by VM* measure reports a value less than 10, then the test will not compute/report the *Retransmit ratio from VM* measure. This is done to ensure that no false alerts are generated by the eG Enterprise system for the *Retransmit ratio from VM* measure. You can change this minimum threshold to any value of your choice.
5. **REPORT BY USER** – For the *Hyper-V* monitoring model, the REPORT BY USER flag is set to **NO** by default, indicating that by default, the guest operating systems on the Hyper-V server are identified using the hostname specified in the operating system. On the other hand, for the *Hyper-V VDI* model, this flag is set to **YES** by default; this implies that in case of VDI servers, by default, the guests will be identified using the login of the user who is accessing the guest OS. In other words, in VDI environments, this test will, by default, report measures for every *username_on_virtualmachinename*.
6. **REPORT POWERED OS** - **This flag becomes relevant only if the REPORT BY USER flag is set to 'Yes'.**

   If the **REPORT POWERED OS** flag is set to **Yes** (which is the default setting), then this test will report measures for even those VMs that do not have any users logged in currently. Such guests will be identified by their *virtualmachine name* and not by the *username_on_virtualmachinename*. On the other hand, if the **REPORT POWERED OS** flag is set to **No**, then this test will not report measures for those VMs to which no users are logged in currently.
7. **IGNORE VMS INSIDE VIEW** - Administrators of some high security Hyper-V environments might not have permissions to internally monitor one/more VMs. The eG agent can be configured to **not obtain the 'inside view' of such 'inaccessible' VMs** using the **IGNORE VMS INSIDE VIEW** parameter. Against this parameter, you can provide a comma-separated list of VM names, or VM name patterns, for which the inside view need not be obtained. For instance, your **IGNORE VMS INSIDE VIEW** specification can be: **xp,*lin*,win*,vista*. Here, the * (asterisk) is used to denote leading and trailing spaces (as the case may be). By default, this parameter is set to *none* indicating that the eG agent

obtains the inside view of all VMs on a Hyper-V host by default.

Note:
While performing VM discovery, the eG agent will not discover the operating system of the VMs configured in the **IGNORE VMS INSIDE VIEW** text box.

8. **EXCLUDE VMS** - Administrators of some virtualized environments may not want to monitor some of their less-critical VMs - for instance, VM templates - both from 'outside' and from 'inside'. The eG agent in this case can be configured to completely exclude such VMs from its monitoring purview. To achieve this, provide a comma-separated list of VMs to be excluded from monitoring in the **EXCLUDE VMS** text box. Instead of VMs, VM name patterns can also be provided here in a comma-separated list. For example, your **EXCLUDE VMS** specification can be: **xp,*lin*,win*,vista*. Here, the * (asterisk) is used to denote leading and trailing spaces (as the case may be). By default, this parameter is set to *none* indicating that the eG agent obtains the inside and outside views of all VMs on a virtual host by default. By providing a comma-separated list of VMs/VM name patterns in the **EXCLUDE VMS** text box, you can make sure the eG agent stops collecting 'inside' and 'outside' view metrics for a configured set of VMs.
9. **IGNORE WINNT** – By default, the eG agent does not support the *inside view* for VMs executing on **Windows NT** operating systems. Accordingly, the **IGNORE WINNT** flag is set to **Yes** by default.

Measurement	Description	Measurement Unit	Interpretation
Segments received by VM	Indicates the rate at which segments are received by the guest.	Segments/Sec	
Segments sent by VM	Indicates the rate at which segments are sent to clients or other guests	Segments/Sec	
Retransmits by VM	Indicates the rate at which segments are being retransmitted by the guest	Segments/Sec	

Measurement	Description	Measurement Unit	Interpretation
Retransmit ratio from VM	Indicates the ratio of the rate of data retransmissions to the rate of data being sent by the guest	Percent	Ideally, the retransmission ratio should be low (< 5%). Most often retransmissions at the TCP layer have significant impact on application performance. Very often a large number of retransmissions are caused by a congested network link, bottlenecks at a router causing buffer/queue overflows, or by lousy network links due to poor physical layer characteristics (e.g., low signal to noise ratio). By tracking the percentage of retransmissions at a guest, an administrator can quickly be alerted to problem situations in the network link(s) to the guest that may be impacting the service performance.

3.7.10 Handles Usage - VM Test

This test monitors and tracks the handles opened by processes running in a target Windows virtual machine.

Target of the test:

A Hyper-V / Hyper-V VDI server

Agent executing the test:

An internal agent

Output of the test:

For a Hyper-V server, one set of results will be reported for every powered-on Windows VM on the server.

For a Hyper-V VDI server, one set of results will be reported for the user who is currently logged into each Windows virtual desktop on the server

Parameters of the test:

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured.
3. **PORT** - The port at which the **HOST** listens. By default, this is NULL.
4. **INSIDE VIEW USING** - By default, this test communicates with every VM remotely and extracts "inside view" metrics. Therefore, by default, the **INSIDE VIEW USING** flag is set to **Remote connection to VM (Windows)**.

   Typically, to establish this remote connection, eG Enterprise requires that the eG agent be configured with domain administrator privileges. In high-security environments, where the IT staff might have reservations about exposing the credentials of their domain administrators, this approach to extracting "inside view" metrics might not be preferred. In such environments therefore, eG Enterprise provides administrators the option to deploy a piece of software called the **eG VM Agent** on every Windows VM; this VM agent allows the eG agent to collect "inside view" metrics from the Windows VMs **without domain administrator rights.** Refer to Section 1.3.2 for more details on the **eG VM Agent**. To ensure that the "inside view" of Windows VMs is obtained using the eG VM Agent, set the **INSIDE VIEW USING** flag to **eG VM Agent (Windows)**. Once this is done, you can set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.
5. **DOMAIN, ADMIN USER, ADMIN PASSWORD,** and **CONFIRM PASSWORD** – By default, this test connects to each virtual guest remotely and attempts to collect "inside view" metrics. Accordingly, the **INSIDE VIEW USING** flag is set to **Remote connection to VM (Windows)** by default. To obtain a remote connection, the test must be configured with the privileges of an administrative user to the domain within which the guests reside. The first step towards this is to specify the DOMAIN within which the virtual guests reside. The **ADMIN USER** and **ADMIN PASSWORD** will change according to the **DOMAIN** specification. Discussed below are the different values that the **DOMAIN** parameter can take, and how they impact the **ADMIN USER** and **ADMIN PASSWORD** specifications:
   - **If the VMs belong to a single domain:** If the guests belong to a specific domain, then specify the name of that domain against the domain parameter. In this case, any administrative user in that domain will have remote access to all the virtual guests. Therefore, an administrator account in the given domain can be provided in the **ADMIN USER** field and the corresponding password in the **ADMIN PASSWORD** field. Confirm the password by retyping it in the **CONFIRM PASSWORD** text box.
   - I**f the VMs belong to different domains:**In this case, you might want to provide multiple domain names. If this is done, then, to access the guests in every configured domain, the test should be configured with the required user privileges; this implies that along with multiple **DOMAIN** names, multiple **ADMIN USER** names and **ADMIN PASSWORDS** would also have to be provided. To help administrators provide these user details quickly and easily, the eG administrative interface embeds a special configuration page. To access this page, simply click on the encircled '+' icon that appears alongside the . To know how to use the special page, refer to Section 3.6.1 of this document.
   - **If the INSIDE VIEW USING flag is set to 'eG VM Agent (Windows)'**: On the other hand, if the **INSIDE VIEW USING** flag is set to **eG VM Agent (Windows)**, then it implies that the inside view can be obtained without domain administrator privileges. Therefore, set the **DOMAIN**, **ADMIN USER**, and

ADMIN PASSWORD parameters to *none*.

4. **REPORT BY USER** – For the *Hyper-V* monitoring model, the REPORT BY USER flag is set to **NO** by default, indicating that by default, the guest operating systems on the Hyper-V server are identified using the hostname specified in the operating system. On the other hand, for the *Hyper-V VDI* model, this flag is set to **YES** by default; this implies that in case of VDI servers, by default, the guests will be identified using the login of the user who is accessing the guest OS. In other words, in VDI environments, this test will, by default, report measures for every *username_on_virtualmachinename*.
5. **REPORT POWERED OS** - **This flag becomes relevant only if the REPORT BY USER flag is set to 'Yes'.**

   If the **REPORT POWERED OS** flag is set to **Yes** (which is the default setting), then this test will report measures for even those VMs that do not have any users logged in currently. Such guests will be identified by their *virtualmachine name* and not by the *username_on_virtualmachinename*. On the other hand, if the **REPORT POWERED OS** flag is set to **No**, then this test will not report measures for those VMs to which no users are logged in currently.
6. **IGNORE VMS INSIDE VIEW** - Administrators of some high security Hyper-V environments might not have permissions to internally monitor one/more VMs. The eG agent can be configured to **not obtain the 'inside view' of such 'inaccessible' VMs** using the **IGNORE VMS INSIDE VIEW** parameter. Against this parameter, you can provide a comma-separated list of VM names, or VM name patterns, for which the inside view need not be obtained. For instance, your **IGNORE VMS INSIDE VIEW** specification can be: **xp,*lin*,win*,vista*. Here, the * (asterisk) is used to denote leading and trailing spaces (as the case may be). By default, this parameter is set to *none* indicating that the eG agent obtains the inside view of all VMs on a Hyper-V host by default.

   Note:
   While performing VM discovery, the eG agent will not discover the operating system of the VMs configured in the **IGNORE VMS INSIDE VIEW** text box.
7. **EXCLUDE VMS** - Administrators of some virtualized environments may not want to monitor some of their less-critical VMs - for instance, VM templates - both from 'outside' and from 'inside'. The eG agent in this case can be configured to completely exclude such VMs from its monitoring purview. To achieve this, provide a comma-separated list of VMs to be excluded from monitoring in the **EXCLUDE VMS** text box. Instead of VMs, VM name patterns can also be provided here in a comma-separated list. For example, your **EXCLUDE VMS** specification can be: **xp,*lin*,win*,vista*. Here, the * (asterisk) is used to denote leading and trailing spaces (as the case may be). By default, this parameter is set to *none* indicating that the eG agent obtains the inside and outside views of all VMs on a virtual host by default. By providing a comma-separated list of VMs/VM name patterns in the **EXCLUDE VMS** text box, you can make sure the eG agent stops collecting 'inside' and 'outside' view metrics for a configured set of VMs.
8. **IGNORE WINNT** – By default, the eG agent does not support the *inside view* for VMs executing on **Windows NT** operating systems. Accordingly, the **IGNORE WINNT** flag is set to **Yes** by default.
9. **HANDLES GROWTH LIMIT** – This defines the upper limit of the handles opened by any process. By default, this parameter is set to 8000.

10. **IGNORE PROCESSES IN DD** - The detailed diagnosis of the *Processes using handles above limit in the VM* measure reveals the top-10 processes in a VM that are using handles above the configured limit, the number of handles used by each process, and the break-up of the handle count by sub-handles (i.e., the count of file handles, disk handles, etc.). For processes that typically open thousands of handles, storing granular, sub-handle-level information pertaining to these handles may impose additional strain on the eG database. In such cases, you can reduce the strain on the eG database by configuring in the **IGNORE PROCESSES IN DD** text box, a comma-separated list of process names/process patterns for which sub-handle-wise breakup need not be collected and stored in the eG database. The default value in this text box is **ccSvcHst.exe**. This implies that, by default, the detailed diagnosis of the *Processes using handles above limit in the VM* measure will only provide the total number of open handles for *ccSvcHst.exe* process, but not the sub-handle-level information. If required, you can choose to exclude the sub-handle-wise breakup from the detailed diagnosis for more processes by including these process names/patterns as part of the **IGNORE PROCESSES IN DD** specification. For instance, your specification can be: **ccSvcHst.exe*,*js.exe*,*java.exe**
11. **DETAILED DIAGNOSIS** - To make diagnosis more efficient and accurate, the eG suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option.

    The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:

    - The eG manager license should allow the detailed diagnosis capability
    - Both the normal and abnormalfrequencies configured for the detailed diagnosis measures should not be 0.

Measurement	Description	Measurement Unit	Interpretation
Handles used by processes of the VM	Indicates the number of handles opened by various processes running in a target Windows virtual machine in the last measurement period.	Number	Use the detailed diagnosis of this measure to determine the top-10 processes in terms of number of handles opened. This information brings to light those processes with too many open handles. By closely tracking the handle usage of these processes over time, you can identify potential handle leaks.

Measurement	Description	Measurement Unit	Interpretation
Processes using handles above limit in the VM	Indicates the number of processes that have opened the handles on or above the value defined in the input parameter - **HANDLES GROWTH LIMIT**.	Number	Using the detailed diagnosis of this measure, you can accurately isolate the process (es) that has opened more handles than the permitted limit. A high value of this measure indicates that too many processes are opening handles excessively. You might want to closely observe the handle usage of these processes over time to figure out whether the spike in usage is sporadic or consistent. A consistent increase in handle usage could indicate a handle leak.

The detailed diagnosis of the *Handles used by processes* measure, if enabled, lists the names of top-10 processes in terms of handle usage, the number of handles each process uses, the process ID, and the ID of the parent process.

List of top 10 processes in a VM that are holding handles

Time	Process Name	Handles used	Process ID	Parent PID
Jan 29, 2009 12:00:49				
	System	3359	0	4
	js	1718	540	6420
	svchost	1208	540	1012
	lsass	1112	492	552
	csrss	1097	420	468
	winlogon	564	420	492
	ImaSrv	559	540	3696
	Rtvscan	536	540	3936
	tomcat	485	540	6572
	services	482	492	540

Figure 3.26: The detailed diagnosis of the Handles used by processes measure

The detailed diagnosis of the *Processes using handles above limit in VM* measure, if enabled, lists the details of processes that are using more handles than the configured limit.

List of processes in a VM that are using handles above the configured handle growth value

Time	Process Name	Handles used	Process ID	Parent PID
Jan 29, 2009 17:54:18	eGRSvc	62410	412	11512

Figure 3.27: The detailed diagnosis of the Processes using handles above limit in VM measure

3.7.11 Windows Services - VM Test

This test tracks the status (whether running or have stopped) of services executing on Windows virtual machines.

Target of the test:

A Hyper-V / Hyper-V VDI server

Agent executing the test:

An internal agent

Output of the test:

For a Hyper-V server, one set of results will be reported for every powered-on Windows VM on the server.

For a Hyper-V VDI server, one set of results will be reported for the user who is currently logged into each Windows virtual desktop on the server

Parameters of the test:

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured.
3. **PORT** - The port at which the **HOST** listens. By default, this is NULL.
4. **INSIDE VIEW USING** - By default, this test communicates with every VM remotely and extracts "inside view" metrics. Therefore, by default, the **INSIDE VIEW USING** flag is set to **Remote connection to VM (Windows)**.

   Typically, to establish this remote connection, eG Enterprise requires that the eG agent be configured with domain administrator privileges. In high-security environments, where the IT staff might have reservations about exposing the credentials of their domain administrators, this approach to extracting "inside view" metrics might not be preferred. In such environments therefore, eG Enterprise provides administrators the option to deploy a piece of software called the **eG VM Agent** on every Windows VM; this VM agent allows the eG agent to collect "inside view" metrics from the Windows VMs **without domain administrator rights.** Refer to Section 1.3.2 for more details on the **eG VM Agent**. To ensure that the "inside view" of Windows VMs is obtained using the eG VM Agent, set the **INSIDE VIEW USING** flag to **eG VM Agent (Windows)**. Once this is done, you can set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.
5. **DOMAIN, ADMIN USER, ADMIN PASSWORD,** and **CONFIRM PASSWORD** – By default, this test connects to each virtual guest remotely and attempts to collect "inside view" metrics. Accordingly, the **INSIDE VIEW USING** flag is set to **Remote connection to VM (Windows)** by default. To obtain a

remote connection, the test must be configured with the privileges of an administrative user to the domain within which the guests reside. The first step towards this is to specify the **DOMAIN** within which the virtual guests reside. The **ADMIN USER** and **ADMIN PASSWORD** will change according to the **DOMAIN** specification. Discussed below are the different values that the **DOMAIN** parameter can take, and how they impact the **ADMIN USER** and **ADMIN PASSWORD** specifications:

- **If the VMs belong to a single domain:** If the guests belong to a specific domain, then specify the name of that domain against the domain parameter. In this case, any administrative user in that domain will have remote access to all the virtual guests. Therefore, an administrator account in the given domain can be provided in the **ADMIN USER** field and the corresponding password in the **ADMIN PASSWORD** field. Confirm the password by retyping it in the **CONFIRM PASSWORD** text box.
- I**f the VMs belong to different domains:**In this case, you might want to provide multiple domain names. If this is done, then, to access the guests in every configured domain, the test should be configured with the required user privileges; this implies that along with multiple **DOMAIN** names, multiple **ADMIN USER** names and **ADMIN PASSWORDS** would also have to be provided. To help administrators provide these user details quickly and easily, the eG administrative interface embeds a special configuration page. To access this page, simply click on the encircled '+' icon that appears alongside the . To know how to use the special page, refer to Section 3.6.1 of this document.
- **If the INSIDE VIEW USING flag is set to 'eG VM Agent (Windows)'**: On the other hand, if the **INSIDE VIEW USING** flag is set to **eG VM Agent (Windows)**, then it implies that the inside view can be obtained without domain administrator privileges. Therefore, set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.

4. **REPORT BY USER** – For the *Hyper-V* monitoring model, the REPORT BY USER flag is set to **NO** by default, indicating that by default, the guest operating systems on the Hyper-V server are identified using the hostname specified in the operating system. On the other hand, for the *Hyper-V VDI* model, this flag is set to **YES** by default; this implies that in case of VDI servers, by default, the guests will be identified using the login of the user who is accessing the guest OS. In other words, in VDI environments, this test will, by default, report measures for every *username_on_virtualmachinename*.
5. **REPORT POWERED OS** - **This flag becomes relevant only if the REPORT BY USER flag is set to 'Yes'.**

   If the **REPORT POWERED OS** flag is set to **Yes** (which is the default setting), then this test will report measures for even those VMs that do not have any users logged in currently. Such guests will be identified by their *virtualmachine name* and not by the *username_on_virtualmachinename*. On the other hand, if the **REPORT POWERED OS** flag is set to **No**, then this test will not report measures for those VMs to which no users are logged in currently.
6. **IGNORE VMS INSIDE VIEW** - Administrators of some high security Hyper-V environments might not have permissions to internally monitor one/more VMs. The eG agent can be configured to **not obtain the 'inside view' of such 'inaccessible' VMs** using the **IGNORE VMS INSIDE VIEW** parameter. Against this parameter, you can provide a comma-separated list of VM names, or VM name patterns, for which the inside view need not be obtained. For instance, your **IGNORE VMS INSIDE VIEW** specification can be: **xp,*lin*,win*,vista*. Here, the * (asterisk) is used to denote leading and trailing

spaces (as the case may be). By default, this parameter is set to *none* indicating that the eG agent obtains the inside view of all VMs on a Hyper-V host by default.

Note:
While performing VM discovery, the eG agent will not discover the operating system of the VMs configured in the **IGNORE VMS INSIDE VIEW** text box.

7. **EXCLUDE VMS** - Administrators of some virtualized environments may not want to monitor some of their less-critical VMs - for instance, VM templates - both from 'outside' and from 'inside'. The eG agent in this case can be configured to completely exclude such VMs from its monitoring purview. To achieve this, provide a comma-separated list of VMs to be excluded from monitoring in the **EXCLUDE VMS** text box. Instead of VMs, VM name patterns can also be provided here in a comma-separated list. For example, your **EXCLUDE VMS** specification can be: **xp,*lin*,win*,vista*. Here, the * (asterisk) is used to denote leading and trailing spaces (as the case may be). By default, this parameter is set to *none* indicating that the eG agent obtains the inside and outside views of all VMs on a virtual host by default. By providing a comma-separated list of VMs/VM name patterns in the **EXCLUDE VMS** text box, you can make sure the eG agent stops collecting 'inside' and 'outside' view metrics for a configured set of VMs.
8. **IGNORE WINNT** – By default, the eG agent does not support the *inside view* for VMs executing on **Windows NT** operating systems. Accordingly, the **IGNORE WINNT** flag is set to **Yes** by default.
9. **IGNORE SERVICES** – Provide a comma-separated list of services that need to be ignored while monitoring. When configuring a service name to exclude, make sure that you specify the **Display Name** of the service, and not the service **Name** you see in the **Services** window on your Windows VM.
10. **IGNORE PROCESSES IN DD** - The detailed diagnosis of the *Processes using handles above limit in the VM* measure reveals the top-10 processes in a VM that are using handles above the configured limit, the number of handles used by each process, and the break-up of the handle count by sub-handles (i.e., the count of file handles, disk handles, etc.). For processes that typically open thousands of handles, storing granular, sub-handle-level information pertaining to these handles may impose additional strain on the eG database. In such cases, you can reduce the strain on the eG database by configuring in the **IGNORE PROCESSES IN DD** text box, a comma-separated list of process names/process patterns for which sub-handle-wise breakup need not be collected and stored in the eG database. The default value in this text box is **ccSvcHst.exe**. This implies that, by default, the detailed diagnosis of the *Processes using handles above limit in the VM* measure will only provide the total number of open handles for *ccSvcHst.exe* process, but not the sub-handle-level information. If required, you can choose to exclude the sub-handle-wise breakup from the detailed diagnosis for more processes by including these process names/patterns as part of the **IGNORE PROCESSES IN DD** specification. For instance, your specification can be: **ccSvcHst.exe*,*js.exe*,*java.exe**
11. **DD FREQUENCY** - Refers to the frequency with which detailed diagnosis measures are to be generated for this test. The default is 1:1. This indicates that, by default, detailed measures will be generated every time this test runs, and also every time the test detects a problem. You can modify this frequency, if you so desire. Also, if you intend to disable the detailed diagnosis capability for this test, you can do so by specifying none against **DD FREQUENCY**.
12. **DETAILED DIAGNOSIS** - To make diagnosis more efficient and accurate, the eG suite embeds an

optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option.

The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:

- The eG manager license should allow the detailed diagnosis capability
- Both the normal and abnormalfrequencies configured for the detailed diagnosis measures should not be 0.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
New automatic services started	Indicates the number of Windows services with startup type as *automatic*, which were running in the last measurement period.	Number	The detailed diagnosis of this measure lists the services (with startup type as *automatic*) that are running.
New automatic services stopped	Indicates the number of Windows services with startup type as *automatic* , which were not running in the last measurement period.	Number	To know which services stopped, use the detailed diagnosis of this measure (if enabled).
New manual services started	Indicates the number of Windows services with startup type as *manual* , which were running in	Number	Use the detailed diagnosis of this measure to identify the services that are running.

Measurement	Description	Measurement Unit	Interpretation
	the last measurement period.		
New manual services stopped	Indicates the number of Windows services with startup type as *manual*, which stopped running in the last measurement period.	Number	To identify the services that stopped, use the detailed diagnosis of this measure.

The detailed diagnosis of the *New automatic services started* measure lists the services that were started recently, the startup type, process ID, and the complete path to the executable that controls the service.

Details of automatic Windows services that have been started recently

Time	Service name	Status	Startup type	ProcessID	Path to executable
Mar 13, 2009 15:22:18					
	Cryptographic Services	Running	Auto	548	C:\Windows\system32 \svchost.exe -k NetworkServic
	DCOM Server Process Launcher	Running	Auto	816	C:\Windows\system32 \svchost.exe -k DcomLaunch
	DHCP Client	Running	Auto	972	C:\Windows\system32 \svchost.exe -k LocalServiceNetworkRestricted
	DNS Client	Running	Auto	548	C:\Windows\system32 \svchost.exe -k NetworkServic
	eGurkhaAgent	Running	Auto	1576	C:\eGurkha\lib\js.exe
	eGVmAgent	Running	Auto	1592	C:\eGVmAgent\lib\js.exe
	Windows Event Log	Running	Auto	972	C:\Windows\System32 \svchost.exe -k LocalServiceNetworkRestricted
	COM+ Event System	Running	Auto	344	C:\Windows\system32 \svchost.exe -k LocalService
	KtmRm for Distributed Transaction Coordinator	Running	Auto	548	C:\Windows\System32 \svchost.exe -k NetworkServic
	Workstation	Running	Auto	344	C:\Windows\System32 \svchost.exe -k LocalService
	TCP/IP NetBIOS Helper	Running	Auto	972	C:\Windows\system32 \svchost.exe -k LocalServiceNetworkRestricted

Figure 3.28: The detailed diagnosis of the New automatic services started measure

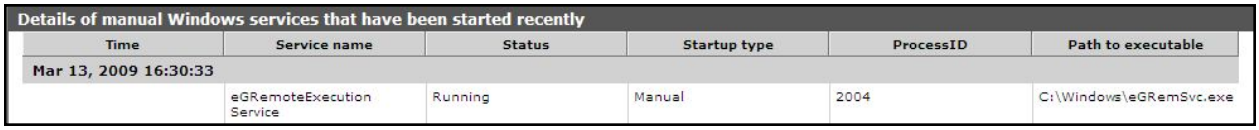
The detailed diagnosis of the *New automatic services stopped* measure lists the services that were stopped recently, the startup type, process ID, and the complete path to the executable that controls the service.

Details of automatic Windows services that have been stopped recently

Time	Service name	Status	Startup type	ProcessID	Path to executable
Mar 13, 2009 10:01:23					
	Background Intelligent Transfer Service	Stopped	Auto	0	C:\Windows\System32 \svchost.exe -k netsvcs
	KtmRm for Distributed Transaction Coordinator	Stopped	Auto	0	C:\Windows\System32 \svchost.exe -k NetworkService
	Distributed Transaction Coordinator	Stopped	Auto	0	C:\Windows\System32 \msdtc.exe
	Windows Remote Management (WS-Management)	Stopped	Auto	0	C:\Windows\System32 \svchost.exe -k NetworkService
	Windows Update	Stopped	Auto	0	C:\Windows\system32 \svchost.exe -k netsvcs

Figure 3.29: The detailed diagnosis of the New automatic services stopped measure

The detailed diagnosis of the *New manual services started* measure lists the services that were started recently, the startup type, process ID, and the complete path to the executable that controls the service.

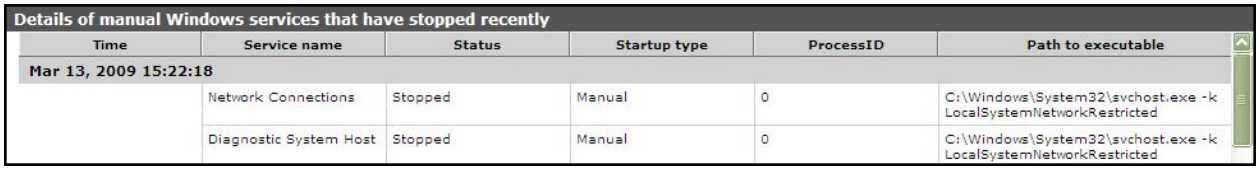


Details of manual Windows services that have been started recently

Time	Service name	Status	Startup type	ProcessID	Path to executable
Mar 13, 2009 16:30:33					
	eGRemoteExecution Service	Running	Manual	2004	C:\Windows\eGRemSvc.exe

Figure 3.30: The detailed diagnosis of the New manual services started measure

The detailed diagnosis of the *New manual services stopped* measure lists the services that were stopped recently, the startup type, process ID, and the complete path to the executable that controls the service.



Details of manual Windows services that have stopped recently

Time	Service name	Status	Startup type	ProcessID	Path to executable
Mar 13, 2009 15:22:18					
	Network Connections	Stopped	Manual	0	C:\Windows\System32\svchost.exe -k LocalSystemNetworkRestricted
	Diagnostic System Host	Stopped	Manual	0	C:\Windows\System32\svchost.exe -k LocalSystemNetworkRestricted

Figure 3.31: The detailed diagnosis of the New manual services stopped measure

3.7.12 Crash Details - VM Test

Event logs on Windows VMs capture critical error conditions such as service crashes and application crashes on the VMs, application and service hangs, and service errors. Since the crash/slowness experienced by any mission-critical program/service on a Windows VM may affect the uptime of the dependent business services, administrators should be able to instantly capture these serious problem conditions, investigate the reasons for their occurrence, and promptly resolve them. This is exactly what the Crash Details -VM test helps administrators achieve! This test periodically scans the event logs on each Windows VM and reports the count of crashes, hangs, and errors that may have occurred recently on that VM. Detailed diagnostics provided by this test pinpoints the applications/services that crashed, hanged, or encountered errors, and thus enables quick and efficient troubleshooting.

Note:

This test will not report metrics on VMs running Windows 2000/2003/XP.

Target of the test: A Hyper-V server

Agent executing the test: An internal agent

Output of the test: One set of results for each Windows VM on a monitored Hyper-V server

Configurable parameters for the test

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured.
3. **PORT** - The port at which the **HOST** listens. By default, this is NULL.
4. **INSIDE VIEW USING** - By default, this test communicates with every VM remotely and extracts “inside view” metrics. Therefore, by default, the **INSIDE VIEW USING** flag is set to **Remote connection to VM**

(Windows).

Typically, to establish this remote connection, eG Enterprise requires that the eG agent be configured with domain administrator privileges. In high-security environments, where the IT staff might have reservations about exposing the credentials of their domain administrators, this approach to extracting "inside view" metrics might not be preferred. In such environments therefore, eG Enterprise provides administrators the option to deploy a piece of software called the **eG VM Agent** on every Windows VM; this VM agent allows the eG agent to collect "inside view" metrics from the Windows VMs **without domain administrator rights.** Refer to Section 1.3.2 for more details on the **eG VM Agent**. To ensure that the "inside view" of Windows VMs is obtained using the eG VM Agent, set the **INSIDE VIEW USING** flag to **eG VM Agent (Windows)**. Once this is done, you can set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.

5. **DOMAIN, ADMIN USER, ADMIN PASSWORD,** and **CONFIRM PASSWORD** – By default, this test connects to each virtual guest remotely and attempts to collect "inside view" metrics. Accordingly, the **INSIDE VIEW USING** flag is set to **Remote connection to VM (Windows)** by default. To obtain a remote connection, the test must be configured with the privileges of an administrative user to the domain within which the guests reside. The first step towards this is to specify the **DOMAIN** within which the virtual guests reside. The **ADMIN USER** and **ADMIN PASSWORD** will change according to the **DOMAIN** specification. Discussed below are the different values that the **DOMAIN** parameter can take, and how they impact the **ADMIN USER** and **ADMIN PASSWORD** specifications:

   - **If the VMs belong to a single domain:** If the guests belong to a specific domain, then specify the name of that domain against the domain parameter. In this case, any administrative user in that domain will have remote access to all the virtual guests. Therefore, an administrator account in the given domain can be provided in the **ADMIN USER** field and the corresponding password in the **ADMIN PASSWORD** field. Confirm the password by retyping it in the **CONFIRM PASSWORD** text box.
   - **If the VMs belong to different domains:**In this case, you might want to provide multiple domain names. If this is done, then, to access the guests in every configured domain, the test should be configured with the required user privileges; this implies that along with multiple **DOMAIN** names, multiple **ADMIN USER** names and **ADMIN PASSWORDS** would also have to be provided. To help administrators provide these user details quickly and easily, the eG administrative interface embeds a special configuration page. To access this page, simply click on the **Click here** hyperlink that appears just above the parameters of this test in the test configuration page. To know how to use the special page, refer to Section 3.7.12 of this document.
   - **If the INSIDE VIEW USING flag is set to 'eG VM Agent (Windows)'**: On the other hand, if the **INSIDE VIEW USING** flag is set to **eG VM Agent (Windows)**, then it implies that the inside view can be obtained without domain administrator privileges. Therefore, set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.

6. **REPORT BY USER** – For the *Microsoft Hyper-V* monitoring model, the **REPORT BY USER** flag is set to **NO** by default, indicating that by default, the guest operating systems on the Hyper-V server are identified using the hostname specified in the operating system. On the other hand, for the *Microsoft Hyper-V VDI* model, this flag is set to **YES** by default; this implies that in case of VDI servers, by

default, the guests will be identified using the login of the user who is accessing the guest OS. In other words, in VDI environments, this test will, by default, report measures for every *username_on_virtualmachinename*.

7. **REPORT POWERED OS** - **This flag becomes relevant only if the REPORT BY USER flag is set to 'Yes'.**

   If the **REPORT POWERED OS** flag is set to **Yes** (which is the default setting), then this test will report measures for even those VMs that do not have any users logged in currently. Such guests will be identified by their *virtualmachine name* and not by the *username_on_virtualmachinename*. On the other hand, if the **REPORT POWERED OS** flag is set to **No**, then this test will not report measures for those VMs to which no users are logged in currently.

8. **REPORT POWERED ON** - You can set the **REPORT POWERED ON** status to **Yes,** so that the test reports an additional measure, *Is VM powered on?*, revealing whether a guest OS is currently running or not. The default status of this flag is set to **Yes** for a *Hyper-V* server. For a *Hyper-V VDI* server on the other hand, the default status of this flag is **No**. This is because, in such environments, the virtual desktops will be in the powered-off state most of the time.

9. **IGNORE VMS INSIDE VIEW** - Administrators of some high security Hyper-V environments might not have permissions to internally monitor one/more VMs. The eG agent can be configured to **not obtain the 'inside view' of such 'inaccessible' VMs** using the **IGNORE VMS INSIDE VIEW** parameter. Against this parameter, you can provide a comma-separated list of VM names, or VM name patterns, for which the inside view need not be obtained. For instance, your **IGNORE VMS INSIDE VIEW** specification can be: **xp,*lin*,win*,vista*. Here, the * (asterisk) is used to denote leading and trailing spaces (as the case may be). By default, this parameter is set to *none* indicating that the eG agent obtains the inside view of all VMs on a Hyper-V host by default.

   Note:

   While performing VM discovery, the eG agent will not discover the operating system of the VMs configured in the **IGNORE VMS INSIDE VIEW** text box.

10. **EXCLUDE VMS** - Administrators of some virtualized environments may not want to monitor some of their less-critical VMs - for instance, VM templates - both from 'outside' and from 'inside'. The eG agent in this case can be configured to completely exclude such VMs from its monitoring purview. To achieve this, provide a comma-separated list of VMs to be excluded from monitoring in the **EXCLUDE VMS** text box. Instead of VMs, VM name patterns can also be provided here in a comma-separated list. For example, your **EXCLUDE VMS** specification can be: **xp,*lin*,win*,vista*. Here, the * (asterisk) is used to denote leading and trailing spaces (as the case may be). By default, this parameter is set to *none* indicating that the eG agent obtains the inside and outside views of all VMs on a virtual host by default. By providing a comma-separated list of VMs/VM name patterns in the **EXCLUDE VMS** text box, you can make sure the eG agent stops collecting 'inside' and 'outside' view metrics for a configured set of VMs.

11. **IGNORE WINNT** – By default, the eG agent does not support the *inside view* for VMs executing on **Windows NT** operating systems. Accordingly, the **IGNORE WINNT** flag is set to **Yes** by default.

12. **DD FREQUENCY** - Refers to the frequency with which detailed diagnosis measures are to be generated for this test. For a *Microsoft Hyper-V* server, this is set to *1:1* by default. This indicates that, by default,

detailed measures will be generated every time this test runs, and also every time the test detects a problem. It is recommended that you do not change the default setting of this parameter. This is because, eG Enterprise can discover the IP addresses of the guest operating systems on a Hyper-V host, only while generating the detailed measures for this test. The automatic discovery of the guest IPs, in turn, enables eG Enterprise to perform **AutoVirtualMapping**.

13. **DETAILED DIAGNOSIS** - To make diagnosis more efficient and accurate, the eG suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option.

    The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:

    - The eG manager license should allow the detailed diagnosis capability
    - Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
Recent application crashes:	Indicates the number of application crash events that occurred on this VM during the last measurement period.	Number	An event with the ID 1000 is logged in the event log every time a program terminates unexpectedly on a virtual desktop. This measure reports the number of events in the event log with event ID 1000. Use the detailed diagnosis of this measure to know which programs and modules stopped suddenly.
Recent service crashes:	Indicates the number of service crash events that occurred on thisVM during the last measurement period.	Number	An event with the ID 7031 is logged in the Service Control Manager every time a service terminates ungracefully. This measure reports the number of events in the event log with event ID 7031. Use the detailed diagnosis of this measure to know the complete details of such events.

Measurement	Description	Measurement Unit	Interpretation
Recent application hangs	Indicates the number of application hang events that occurred on this VM during the last measurement period.	Number	An event with the ID 1002 is logged in the Application Event Log every time an application hangs. This measure reports the number of events in the event log with event ID 1002. Use the detailed diagnosis of this measure to know the complete details of the recent application hang events.
Recent service hangs:	Indicates the number of service hang events that occurred on this VM during the last measurement period.	Number	An event with the ID 7022 is logged in the Service Control Manager every time a service hangs. This measure reports the number of events in the event log with event ID 7022. Use the detailed diagnosis of this measure to know the complete details of the recent service hang events.
Recent service errors:	Indicates the number of service errors that occurred on this VM during the last measurement period.	Number	Events with the ID 7023, 7024, and 7026 are logged in the Service Control Manager every time a service error occurs. This measure reports the number of events in the event log with the aforesaid event IDs. Use the detailed diagnosis of this measure to know the complete details of the recent service errors.

3.8 Correlation Between Applications in a Hyper-V Virtualized Environment

Using the eG Enterprise administration console, administrators can add applications running on the VMs for monitoring. To monitor these applications, agents can be installed in the guests, or an agentless monitoring approach can be used. To effectively monitor the applications running in a virtual environment, it is important to be able to determine which *Hyper-V* server an application is running on. This mapping of applications to virtual servers is important for root-cause diagnosis – for example, a problem with the virtual server (e.g., excessive disk slowdowns) can impact the performance of all the applications running on the server's virtual machines.

eG Enterprise is able to automatically determine the mapping of applications to virtual servers. Whether eG Enterprise automatically determines the mapping of applications to virtual servers or not is determined by the value of the **AutoVirtualMapping** variable in the **[MISC]** section of the **eg_external.ini** configuration file in the **<EG_INSTALL_DIR>\manager\config** directory of the eG manager. If the value of this variable is **true,** the eG manager auto-discovers the applications to virtual servers mapping.

Note:

- For **AutoVirtualMapping** to work, the detailed diagnosis frequencies set globally (i.e., using the Configure -> Diagnosis menu sequence) should not be set to *0:0*.
- As long as the **Identify agents only using nick names** flag in the **MANAGER SETTINGS** page of the eG administrative interface (Configure -> Manager Settings menu sequence) is **Yes** (which is the default), eG Enterprise can automatically identify the server applications executing on a Hyper-V host, using the host/nick names that are mapped to the IP addresses discovered on the host. If the **Identify agents only using nick names** flag is set to **No** instead, then make sure that, while managing a server application executing in a virtualized environment, the hostname of the virtual machine is specified as the nick name of the corresponding server application. If more than one server application is executing on the same virtual machine, then any one of those server applications should have the virtual machine name as its nick name.

To disable auto-discovery, set this value to **false.** In such a case, once a *Microsoft Hyper-V* server is added, then, when adding any new server application using the eG administrative interface, you will be prompted to manually set an association between the server application being added and the *Virtual Server*.

The mapping of applications to virtual servers is used by eG Enterprise for correlation – e.g., since the application runs on the virtual server, it is most likely that a problem with the virtual server will impact the performance of the application running on one of the guests. To view this application-virtual server association, simply click on the **VIRTUAL TOPOLOGY** link in the layer model page of the virtual server.

Note:

The **VIRTUAL TOPOLOGY** link will also be available in the layer model page of those server applications that are executing on virtual guests.

Doing so reveals Figure 3.32 depicting the *Hyper-V* server and the server applications executing on it. By clicking on any of the components in Figure 3.32, the user can drill down into specific layers of this component for specific details on the performance of the component.

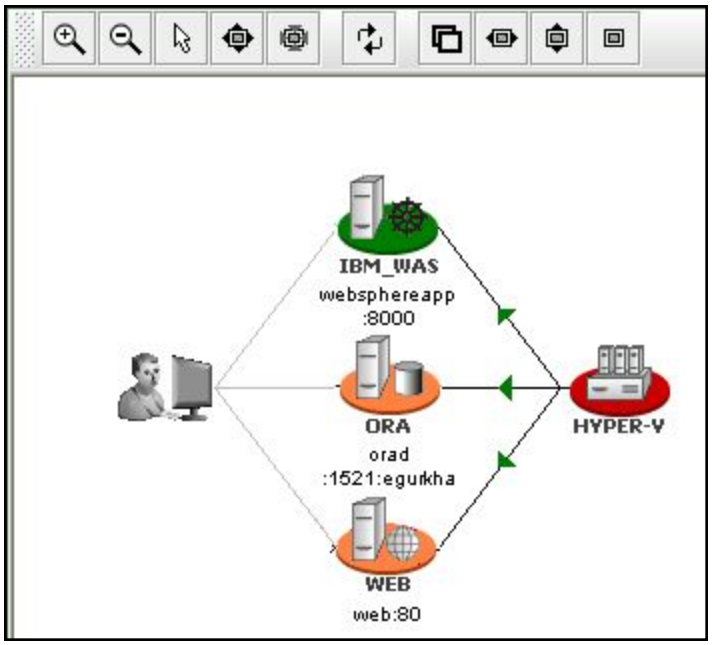


Figure 3.32: Depicts the applications that have been deployed on the guest OS of a virtual server

The arrows in Figure 3.32 depict the dependencies between the virtual server host and the applications running on it. Since the applications are hosted on one of the guests running on the host, they depend on the virtual server host – i.e., any unusual resource usage on the virtual server host impacts the applications running on any of the virtual guests. The dependency information between the virtual server host and the applications hosted on it is used by eG Enterprise for end-to-end correlation.

3.9 Troubleshooting

3.9.1 Troubleshooting the Failure of the eG Agent to Auto-discover the IP Addresses of VMs

If the eG agent is not able to discover the IP addresses of one/more VMs, then follow the steps given below:

a. Login to the root partition.

Go to the command prompt.

Switch to the directory, **<EG_INSTALL_DIR>\lib**.

Run the following command from that directory:

cscript eG_HypervGuestInfo.vbs

If this script executes successfully, it would return the following information:

- The names of the VMs on the Hyper-V host;
    - The current state of each VM (the value *2* if the VM is running, and the value *3* if it is powered-off);
- The Fully Qualified Domain Name (FQDN) of every VM

- The operating system on which each VM is executing

Make a note of the FQDN of a VM.

Then, issue the following command at the command prompt to identify the IP address of that VM:

nslookup <FQDN>

If this command fails to return the IP address of the VM, it could mean that the IP address is not resolvable in the DNS server.

On the other hand, if the **eG_HypervGuestInfo.vbs** script itself fails to return the FQDN and the operating system of the VMs, it could indicate one/more of the following:

- For VM discovery to occur, the **Integration Services** component should be installed on every target VM. Non-availability of the **Integration Services** component on a VM could cause the script to not report the FQDN and operating system of that VM.

- The script may also fail if the target VMs are executing on any of the following Windows operating systems:
  - Windows Server 2008 64-bit
  - Windows Server 2008 x86
  - Windows Server 2003 x64 with SP2
  - Windows 2000 Server with SP4
  - Windows 2000 Advanced Server SP4
  - Windows Vista x64 with SP1
  - Windows Vista x86 with SP1
  - Windows XP x86 with SP2/SP3
  - Windows XP x64 with SP2

This is because the **eG_HypervGuestInfo.vbs** script is not supported on any of the above-mentioned opeating systems.

17. **Similarly, if a VM is executing on a Linux operating system, then again the eG_HypervGuestInfo.vbs** script will not be able to retrieve the FQDN of that VM.

Chapter

4

The Hyper-V VDI Monitoring Model

In some environments, the virtual guests hosted on Hyper-V servers may be used to support desktop applications. Administrators of such virtual environments would want to know the following:

- How many desktops are powered on simultaneously on the Hyper-V Server?
- Which users are logged on and when did each user login?
- How much CPU, memory, disk and network resources is each desktop taking?
- What is the typical duration of a user session?
- Who has the peak usage times?
- What applications are running on each desktop?
- Which Hyper-V server is a virtual guest running on?
- When was a guest moved from a Hyper-V Server? Which server was the guest moved to?
- Why was the guest migrated? What activities on the Hyper-V host caused the migration?

Using the *Hyper-V VDI* model (see Figure 4.1), administrators can find quick and accurate answers to all the queries above, and also receive a complete 'desktop view', which allows them to get up, close with the performance of every guest OS hosted by the Hyper-V server and detect anomalies (if any) in its functioning.

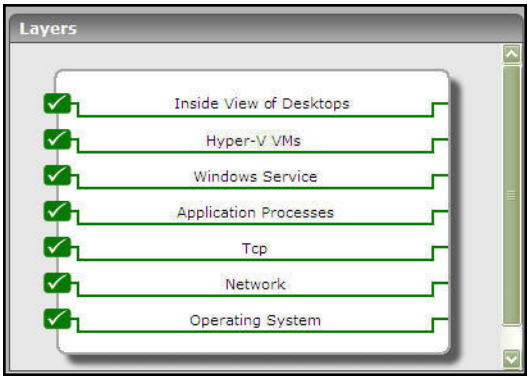


Figure 4.1: Figure 3.1: The layer model of a Hyper-V VDI server

The layers depicted by Figure 4.1 and the tests associated with the layers are discussed in detail in the sections that follow. Since the last 4 layers of the model have already been dealt with in the previous section, this section will discuss the **Hyper-V VMs** and the **Inside View of Desktops** layer only.

4.1 The Outside View of VMs Layer

The **Outside View of VMs** layer provides the host operating system's view of the resource usage levels of each of the virtual guests hosted on it. Using the information reported by this test, administrators can:

- Determine which of the guests is taking up more resources (CPU, memory, network, or disk) than the others. This information can help with load balancing or capacity planning. For example, if one of the guests is receiving a very high rate of requests compared to the others, this guest may be a candidate for migration to another ESX server, so as to minimize the impact it has on the other guests on the current server.
- Determine times when sudden or steady spikes in the physical resource utilization are caused by the guest machines
- Know which guest systems at what times experienced heavy session loads or unexpected session logouts

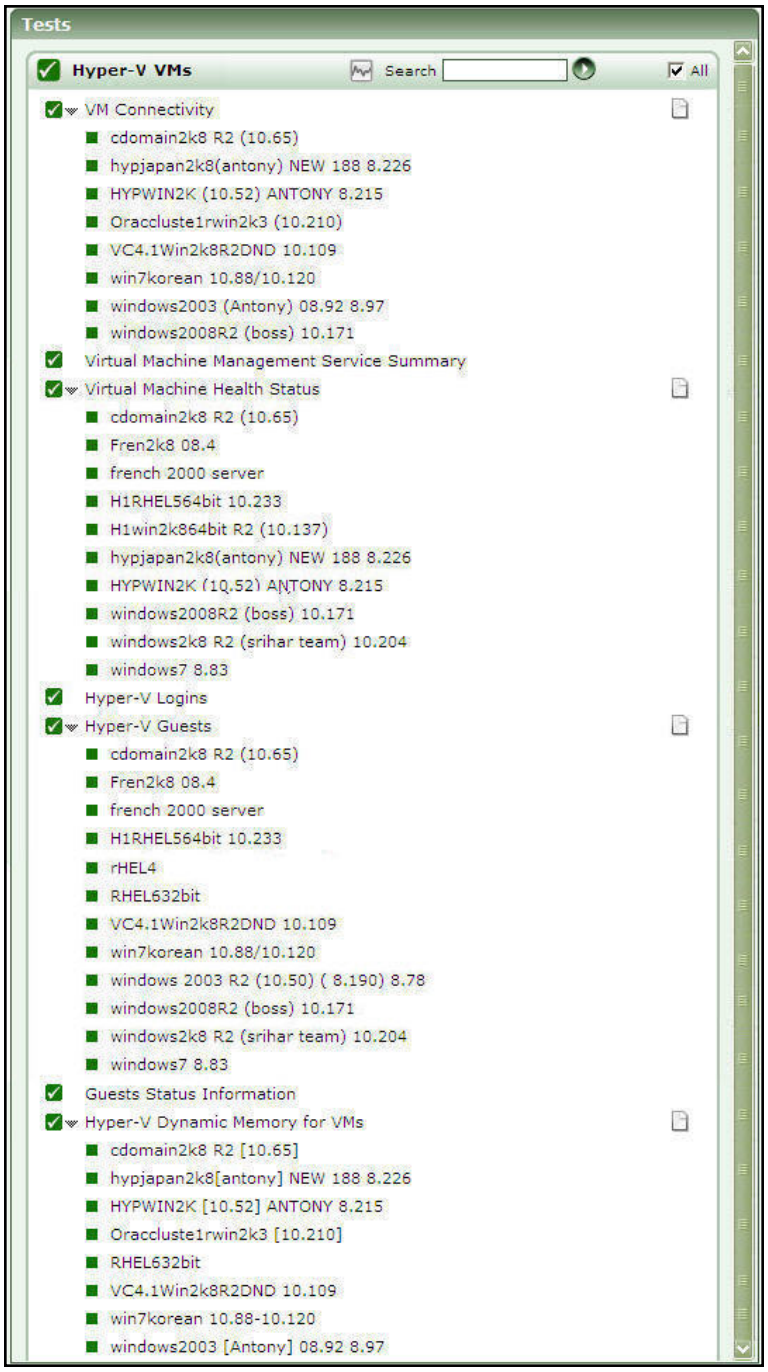


Figure 4.2: Figure 3.2: The tests associated with the Outside View of VMs layer

The **VM Connectivity** test, and the **Virtual Machine Management Service Summary** test have already been discussed in Chapter 2 of this document. Since the **Hyper-V Guests** test and **Guests Status Information** test reports additional measures for the *Hyper-V VDI* model, and because the **Hyper-V Logins** test applies only to this model, the sections to come discuss these 3 tests alone in detail.

4.1.1 Hyper-V VM Details Test

This test monitors the amount of the physical server's resources that each guest on a Hyper-V server is taking up. Using the metrics reported by this test, administrators can determine which virtual guest is taking up most CPU, which guest is generating the most network traffic, which guest is taking up the maximum memory utilization, which guest has the maximum disk activity, which disk has the maximum number of user sessions etc.

Target of the test	**A Hyper-V server**
Agent deploying the test	**An internal agent**

Configurable parameters for the test	1. **TESTPERIOD** - How often should the test be executed **HOST** - The host for which the test is to be configured. **PORT** - The port at which the **HOST** listens. By default, this is NULL. **INSIDE VIEW USING** - By default, this test communicates with every VM remotely and extracts "inside view" metrics. Therefore, by default, the **INSIDE VIEW USING** flag is set to **Remote connection to VM (Windows)**. Typically, to establish this remote connection, eG Enterprise requires that the eG agent be configured with domain administrator privileges. In high-security environments, where the IT staff might have reservations about exposing the credentials of their domain administrators, this approach to extracting "inside view" metrics might not be preferred. In such environments therefore, eG Enterprise provides administrators the option to deploy a piece of software called the **eG VM Agent** on every Windows VM; this VM agent allows the eG agent to collect "inside view" metrics from the Windows VMs **without domain administrator rights.** Refer to Section 1.3.2 for more details on the **eG VM Agent**. To ensure that the "inside view" of Windows VMs is obtained using the eG VM Agent, set the **INSIDE VIEW USING** flag to **eG VM Agent (Windows)**. Once this is done, you can set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*. **DOMAIN, ADMIN USER, ADMIN PASSWORD,** and **CONFIRM PASSWORD** – By default, this test connects to each virtual guest remotely and attempts to collect "inside view" metrics. Accordingly, the **INSIDE VIEW USING** flag is set to **Remote connection to VM (Windows)** by default. To obtain a remote connection, the test must be configured with the privileges of an administrative user to the domain within which the guests reside. The first step towards this is to specify the DOMAIN within which the virtual guests reside. The **ADMIN USER** and **ADMIN PASSWORD** will change according to the **DOMAIN** specification. Discussed below are the different values that the **DOMAIN** parameter can take, and how they impact the **ADMIN USER** and **ADMIN PASSWORD** specifications:

	• **If the VMs belong to a single domain**: **If the guests belong to a specific domain, then specify the name of that domain against the DOMAIN** parameter. In this case, any **administrative user in that domain will have remote access to all the virtual guests. Therefore, an administrator account in the given domain can be provided in the** ADMIN USER **field and the corresponding password in the** ADMIN PASSWORD **field. Confirm the password by retyping it in the** CONFIR**M PASSWORD text box.** • If the VMs belong to different domains: In this case, you might want to provide multiple domain names. If this is done, then, to access the guests in every configured domain, the test should be configured with the required user privileges; this implies that along with multiple **DOMAIN** names, multiple **ADMIN USER** names and **ADMIN PASSWORDS** would also have to be provided. To help administrators provide these user details quickly and easily, the eG administrative interface embeds a special configuration page. To access this page, simply click on the **Click here** hyperlink that appears just above the parameters of this test in the test configuration page. To know how to use the special page, refer to Section 3.6.1 of this document.
	• **If the INSIDE VIEW USING flag is set to 'eG VM Agent (Windows)'**: On the other hand, if the **INSIDE VIEW USING** flag is set to **eG VM Agent (Windows)**, then it implies that the inside view can be obtained without domain administrator privileges. Therefore, set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*. **REPORT BY USER** – For the *Hyper-V* monitoring model, the REPORT BY USER flag is set to **NO** by default, indicating that by default, the guest operating systems on the Hyper-V server are identified using the hostname specified in the operating system. On the other hand, for the *Hyper-V VDI* model, this flag is set to **YES** by default; this implies that in case of VDI servers, by default, the guests will be identified using the login of the user who is accessing the guest OS. In other words, in VDI environments, this test will, by default, report measures for every *username_on_virtualmachinename*. **AGGREGATE USER SESSIONS** –This flag is closely related to the **REPORT BY USER** flag. Since the **REPORT BY USER** flag is set to **No** by default for a VMware ESX server, this test will, by default, ignore the status of the **AGGREGATE USER SESSIONS** flag while monitoring that server. In case of the VDI model on the other hand, the **REPORT BY USER** flag is set to **Yes** by default. Therefore, the status of the **AGGREGATE USER**

	SESSIONS flag gains significance in the case of the VDI server. By default, the **AGGREGATE USER SESSIONS** flag is set to **No**. This implies that if a single user is currently logged into multiple guests, then this test, by default, will report a set of measures for every *username on guestname*. On the other hand, if the status of this flag is changed to **Yes**, then, this test will report a set of (aggregated) measures for every distinct *user* to the virtual desktop environment. In other words, this test will report measures that are aggregated across all the currently active sessions for a user, spanning multiple VMs. **REPORT POWERED OS** - **This flag becomes relevant only if the REPORT BY USER flag is set to 'Yes'.** If the **REPORT POWERED OS** flag is set to **Yes** (which is the default setting), then this test will report measures for even those VMs that do not have any users logged in currently. Such guests will be identified by their *virtual machine name* and not by the *username_on_virtualmachinename*. On the other hand, if the **REPORT POWERED OS** flag is set to **No**, then this test will not report measures for those VMs to which no users are logged in currently. **REPORT POWERED ON** - You can set the REPORT POWERED ON status to **YES**, so that the test reports an additional measure, *Is VM powered on?*, revealing whether a guest OS is currently running or not. The default status of this flag is set to **Yes** for a *Hyper-V* server. For a *Hyper-V VDI* server on the other hand, the default status of this flag is **No**. This is because, in such environments, the virtual desktops will be in the powered-off state most of the time.
	IGNORE VMS INSIDE VIEW - Administrators of some high security Hyper-V environments might not have permissions to internally monitor one/more VMs. The eG agent can be configured to **not obtain the 'inside view' of such 'inaccessible' VMs** using the **IGNORE VMS INSIDE VIEW** parameter. Against this parameter, you can provide a comma-separated list of VM names, or VM name patterns, for which the inside view need not be obtained. For instance, your **IGNORE VMS INSIDE VIEW** specification can be: **xp,*lin*,win*,vista*. Here, the * (asterisk) is used to denote leading and trailing spaces (as the case may be). By default, this parameter is set to *none* indicating that the eG agent obtains the inside view of all VMs on a Hyper-V host by default. **Note:** While performing VM discovery, the eG agent will not discover the operating system of the VMs configured in the **IGNORE VMS INSIDE VIEW** text box. **EXCLUDE VMS** - Administrators of some virtualized environments may not want to monitor

	some of their less-critical VMs - for instance, VM templates - both from 'outside' and from 'inside'. The eG agent in this case can be configured to completely exclude such VMs from its monitoring purview. To achieve this, provide a comma-separated list of VMs to be excluded from monitoring in the **EXCLUDE VMS** text box. Instead of VMs, VM name patterns can also be provided here in a comma-separated list. For example, your **EXCLUDE VMS** specification can be: **xp,*lin*,win*,vista*. Here, the * (asterisk) is used to denote leading and trailing spaces (as the case may be). By default, this parameter is set to *none* indicating that the eG agent obtains the inside and outside views of all VMs on a virtual host by default. By providing a comma-separated list of VMs/VM name patterns in the **EXCLUDE VMS** text box, you can make sure the eG agent stops collecting 'inside' and 'outside' view metrics for a configured set of VMs. **IGNORE WINNT** – By default, the eG agent does not support the *inside view* for VMs executing on **Windows NT** operating systems. Accordingly, the **IGNORE WINNT** flag is set to **Yes** by default. **DETAILED DIAGNOSIS** - To make diagnosis more efficient and accurate, the eG suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option. The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled: • The eG manager license should allow the detailed diagnosis capability • Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.
Outputs of the test	**One set of results for every** *user* or *guest* or *useronguest* **to the Hyper-V server monitored**

Measurements made by the test	Measurement	Measurement Unit	Interpretation

<table>
<tr>
<td></td>
<td>Current sessions:
This measure is relevant only for monitoring of virtual desktops (i.e., for Hyper-V VDI servers). When reporting metrics for specific users, this metric indicates the number of sessions that each user has currently logged into; this measure will be available only if the test reports measures per currently logged in user.</td>
<td>Number</td>
<td>This is a good indicator of how busy the user is. The detailed diagnosis of this measure, if enabled, reveals the guests to which the user is currently logged on to.</td>
</tr>
<tr>
<td></td>
<td>Is VM powered on?:
Whether the virtual machine is Hyper-V server host or not.</td>
<td></td>
<td>While the test reports a wide variety of other metrics too for virtual machines that are alive, only the Powered on status is indicated for virtual machines that are currently not available.

If this measure reports the value On, it indicates that the guest is up and running. The value Off could indicate that the guest has been powered- off; it could also indicate that the guest has moved to a different Hyper-V server.

The numeric values that correspond to each of the powered-on states discussed above are listed in the table below:
<table><tr><th>State</th><th>Value</th></tr><tr><td>On</td><td>1</td></tr><tr><td>Off</td><td>0</td></tr></table>
Note:
By default, this measure reports the values On or Off to indicate the status of a VM. The graph of this measure however, represents the status of a VM using the numeric equivalents - 0 or 1.</td>
</tr>
<tr>
<td></td>
<td>Virtual CPU allocated to VM:
Indicates the number of processors present in this VM.</td>
<td>Number</td>
<td>All execution in the root and child partitions (where guest VMs run) happens on Virtual Processors (VPs). At a minimum, you will see one VP for each Logical Processor (LP). These account for the root VPs. You will then see one for</td>
</tr>
</table>

			each VP you have configured to a guest. Therefore, if you have an 8LP system with 1 guest running with 2 VPs, the count here will be 10.
	Virtual CPU Utilization of VM: ***Indicates the*** **percentage of time spent by the virtual processor assigned to this VM in guest and hypervisor code**.	**Percent**	This measure serves as an effective indicator of how resource-intensive a particular VM is on a specific Hyper-V server.
	Virtual machine runtime: Indicates the **percentage of time spent by the virtual processor in guest code.**	**Percent**	Comparing the value of the *Virtual machine runtime* and *Hypervisor runtime* measures for every VM will reveal where the virtual processors of the VM have spent more time – in processing guest code or in processing hypervisor code?
	Hypervisor runtime: Indicates the percentage of time the virtual processor spend in hypervisor code.	**Percent**	
	Memory allocated to VM: Indicates the amount of physical memory allocated to this VM.	**MB**	
	Data transmitted by VM: Indicates the number of bytes per second sent over the network adapters supported by this VM.	**Mbps**	
	Data received by VM: Indicates the number of bytes per second sent over the network adapters supported by this VM.	**Mbps**	
	Data dropped by VM: Indicates the number of bytes dropped on the network adapter.	**MB**	Ideally, this value should be very low. A high value could be indicative of a network bottleneck.
	Disk reads by VM: Indicates the **number of bytes read per second from the disks attached to the IDE controller.**	**MB/Sec**	These measures are good indicators of the activity on the disks attached to the IDE controller.
	Disk writes by VM:	**MB/Sec**	

	Indicates the **the number of bytes written per second to the disks attached to the IDE controller.**		
	Deposited pages: Indicates the number of memory pages deposited into the partition.	**Number**	**For each partition, the hypervisor maintains a memory pool of RAM SPA pages. This pool acts just like a checking account. The amount of pages in the pool is called the** *balance*. Pages are *deposited* or *withdrawn* from the pool. When a hypercall that requires memory is made by a partition, the hypervisor withdraws the required memory from the total pool balance of that partition. When the *balance* available in the pool is less, then more memory pages are *deposited* in the pool. A very high value of this measure therefore, indicates that the *balance* in the pool maintained for this partition is dwindling. This is a cause for concern.
	Hypercalls: Indicates the **rate of hypercalls made by this guest's code on the virtual processor.**	**Hypercalls/Sec**	**Hypercalls are one form of** *enlightenment*. **Guest OS's use the enlightenments to more efficiently use the system via the hypervisor. TLB flush is an example hypercall. If this value is zero, it is an indication that Integration Components are not installed. New OS's like WS08 can use hypercalls without enlightened drivers. So, hypercalls are only a prerequisite and not a guarantee for not having Integration Components installed.**
	Control register accesses: Indicates the **rate of control register accesses by this guest on its virtual processors.**	**Accesses/Sec**	Control registers are used to set up address mapping, privilege mode, etc.
	HLT instructions:	**Instructions/Sec**	A HLT will cause the hypervisor scheduler

	Indicates the **rate of HLT instructions executed by this guest on its virtual processors.**		to de-schedule the current VP and move to the next VP in the runlist.
	Emulated instruction: Indicates the **rate of emulated instructions while executing guest code on the virtual processor.**	**Instructions/Sec**	
	MWAIT instructions: Indicates the rate of MWAIT instructions executed by this guest on its virtual processors.	**Instructions/Sec**	**The** mwait **(monitored wait) instruction instructs the processor to enter a wait state in which the processor is instructed to monitor the address range between** a **and** b **and wait for an event or a store to that address range.**
	CPUID instructions: Indicates the rate of CPUID instructions executed by this guest on its virtual processors.	**Instructions/Sec**	The CPUID instruction is used to retrieve information on the local CPU's capabilities. Typically, CPUID is only called when the OS / Application first start. Therefore, this value is likely to be 0 most of the time.
	Page fault intercepts: **Indicates the rate of page fault exceptions intercepted by the hypervisor while executing this guest's code on the virtual processor**	**Intercepts/Sec**	**Whenever guest code accesses a page not in the CPU TLB a page fault will occur. This counter is closely correlated with** the *Large Page TLB Fills* **measure.**
	Total intercepts : Indicates the rate of hypervisor intercept messages.	**Intercepts/Sec**	Whenever a guest VP needs to exit its current mode of running for servicing in the hypervisor, this is called an intercept. Some common causes of intercepts are resolving Guest Physical Address (GPA) to Server Physical Address (SPA) translations, privileged instructions like hlt / cupid / in / out, and the end of the VP's scheduled time slice.
	Large page TLB fills: Indicates the rate of virtual TLB fills on large pages.	**Fills/Sec**	There are two types of TLB entries (and some three). Small TLB which generally means a 4K page and large Page which generally means 2MB. There are fewer Large TLB entries on the order of 8 – 32. **A non-zero value for this**

			measures indicates that the root partition is using large pages.
	Small page TLB fills: Indicates the rate of virtual TLB fills on 4K pages.	**Fills/Sec**	There are two types of TLB entries (and some three). Small TLB which generally means a 4K page and large Page which generally means 2MB. There are fewer Large TLB entries on the order of 64 – 1024+.
	Cpu utilization of VM: Indicates the percentage of allocated CPU resources that this VM is currently using.	**Percent**	Comparing the value of this measure across VMs will enable you to accurately identify the VMs on which CPU-intensive applications are executing.

The detailed diagnosis of the *Current sessions* measure reveals the guests to which the user is currently logged in.

Details of current user sessions

Time	GuestName	UserName	OS
Mar 24, 2009 12:19:41	win200864bit	win-2008xb64\administrator	Windows Server (R) 2008 Standard

Figure 4.3: Figure 3.3: The detailed diagnosis of the Current sessions measure

4.1.2 Hyper-V VM Information Test

Hyper-V™ live migration is designed to move running VMs with no impact on VM availability to users. By pre-copying the memory of the migrating VM to the destination physical host, live migration minimizes the amount of transfer time of the VM A live migration is deterministic, meaning that the administrator, or script, that initiates the live migration can control which computer is the destination for the live migration. The guest operating system in the migrating VM is unaware that the migration is happening, so no special configuration for the guest operating system is needed.

Below is a summary of the live migration process:

- All VM memory pages are transferred from the source Hyper-V™ physical host to the destination Hyper-V™ physical host. While this is occurring, any VM modifications to its memory pages are tracked.
- ™Pages that were modified while step 1 was occurring are transferred to the destination physical computer.
- The storage handle for the VM's VHD files are moved to the destination physical computer.
- The destination VM is brought online on the destination Hyper-V™ server.

This test reports the number of guests registered with the server, and promptly alerts administrators to addition/removal of guests from the server.

Target of the test:

A Hyper-V server hosting virtual desktops

Agent executing the test:

An internal agent

Output of the test

One set of results for the Hyper-V server monitored

Parameters of the test:

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured.
3. **PORT** - The port at which the **HOST** listens. By default, this is NULL.
4. **INSIDE VIEW USING** - By default, this test communicates with every VM remotely and extracts "inside view" metrics. Therefore, by default, the **INSIDE VIEW USING** flag is set to **Remote connection to VM (Windows)**.

   Typically, to establish this remote connection, eG Enterprise requires that the eG agent be configured with domain administrator privileges. In high-security environments, where the IT staff might have reservations about exposing the credentials of their domain administrators, this approach to extracting "inside view" metrics might not be preferred. In such environments therefore, eG Enterprise provides administrators the option to deploy a piece of software called the **eG VM Agent** on every Windows VM; this VM agent allows the eG agent to collect "inside view" metrics from the Windows VMs **without domain administrator rights.** Refer to Section 1.3.2 for more details on the **eG VM Agent**. To ensure that the "inside view" of Windows VMs is obtained using the eG VM Agent, set the **INSIDE VIEW USING** flag to **eG VM Agent (Windows)**. Once this is done, you can set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.
5. **DOMAIN, ADMIN USER, ADMIN PASSWORD,** and **CONFIRM PASSWORD** – By default, this test connects to each virtual guest remotely and attempts to collect "inside view" metrics. Accordingly, the **INSIDE VIEW USING** flag is set to **Remote connection to VM (Windows)** by default. To obtain a remote connection, the test must be configured with the privileges of an administrative user to the domain within which the guests reside. The first step towards this is to specify the **DOMAIN** within which the virtual guests reside. The **ADMIN USER** and **ADMIN PASSWORD** will change according to the **DOMAIN** specification. Discussed below are the different values that the **DOMAIN** parameter can take, and how they impact the **ADMIN USER** and **ADMIN PASSWORD** specifications:

- **If the VMs belong to a single domain:** If the guests belong to a specific domain, then specify the name of that domain against the domain parameter. In this case, any administrative user in that domain will have remote access to all the virtual guests. Therefore, an administrator account in the given domain can be provided in the **ADMIN USER** field and the corresponding password in the **ADMIN PASSWORD** field. Confirm the password by retyping it in the **CONFIRM PASSWORD** text box.
- **If the VMs belong to different domains:**In this case, you might want to provide multiple domain names. If this is done, then, to access the guests in every configured domain, the test should be configured with the required user privileges; this implies that along with multiple **DOMAIN** names, multiple **ADMIN USER** names and **ADMIN PASSWORDS** would also have to be provided. To help administrators provide these user details quickly and easily, the eG administrative interface embeds a special configuration page. To access this page, simply click on the **Click here** hyperlink that appears just above the parameters of this test in the test configuration page. To know how to use the special page, refer to Section 4.1.2 of this document.
- **If the INSIDE VIEW USING flag is set to 'eG VM Agent (Windows)':** On the other hand, if the **INSIDE VIEW USING** flag is set to **eG VM Agent (Windows)**, then it implies that the inside view can be obtained without domain administrator privileges. Therefore, set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.

6. **REPORT BY USER** – For the *Microsoft Hyper-V* monitoring model, the **REPORT BY USER** flag is set to **NO** by default, indicating that by default, the guest operating systems on the Hyper-V server are identified using the hostname specified in the operating system. On the other hand, for the *Microsoft Hyper-V VDI* model, this flag is set to **YES** by default; this implies that in case of VDI servers, by default, the guests will be identified using the login of the user who is accessing the guest OS. In other words, in VDI environments, this test will, by default, report measures for every *username_on_ virtualmachinename*.
7. **REPORT POWERED OS** - **This flag becomes relevant only if the REPORT BY USER flag is set to 'Yes'.**

   If the **REPORT POWERED OS** flag is set to **Yes** (which is the default setting), then this test will report measures for even those VMs that do not have any users logged in currently. Such guests will be identified by their *virtualmachine name* and not by the *username_on_virtualmachinename*. On the other hand, if the **REPORT POWERED OS** flag is set to **No**, then this test will not report measures for those VMs to which no users are logged in currently.
8. **REPORT POWERED ON** - You can set the **REPORT POWERED ON** status to **Yes,** so that the test reports an additional measure, *Is VM powered on?*, revealing whether a guest OS is currently running or not. The default status of this flag is set to **Yes** for a *Hyper-V* server. For a *Hyper-V VDI* server on the other hand, the default status of this flag is **No**. This is because, in such environments, the virtual desktops will be in the powered-off state most of the time.
9. **IGNORE VMS INSIDE VIEW** - Administrators of some high security Hyper-V environments might not have permissions to internally monitor one/more VMs. The eG agent can be configured to **not obtain the 'inside view' of such 'inaccessible' VMs** using the **IGNORE VMS INSIDE VIEW** parameter. Against this parameter, you can provide a comma-separated list of VM names, or VM name patterns,

for which the inside view need not be obtained. For instance, your **IGNORE VMS INSIDE VIEW** specification can be: **xp,*lin*,win*,vista*. Here, the * (asterisk) is used to denote leading and trailing spaces (as the case may be). By default, this parameter is set to *none* indicating that the eG agent obtains the inside view of all VMs on a Hyper-V host by default.

Note:

While performing VM discovery, the eG agent will not discover the operating system of the VMs configured in the **IGNORE VMS INSIDE VIEW** text box.

10. **EXCLUDE VMS** - Administrators of some virtualized environments may not want to monitor some of their less-critical VMs - for instance, VM templates - both from 'outside' and from 'inside'. The eG agent in this case can be configured to completely exclude such VMs from its monitoring purview. To achieve this, provide a comma-separated list of VMs to be excluded from monitoring in the **EXCLUDE VMS** text box. Instead of VMs, VM name patterns can also be provided here in a comma-separated list. For example, your **EXCLUDE VMS** specification can be: **xp,*lin*,win*,vista*. Here, the * (asterisk) is used to denote leading and trailing spaces (as the case may be). By default, this parameter is set to *none* indicating that the eG agent obtains the inside and outside views of all VMs on a virtual host by default. By providing a comma-separated list of VMs/VM name patterns in the **EXCLUDE VMS** text box, you can make sure the eG agent stops collecting 'inside' and 'outside' view metrics for a configured set of VMs.
11. **IGNORE WINNT** – By default, the eG agent does not support the *inside view* for VMs executing on **Windows NT** operating systems. Accordingly, the **IGNORE WINNT** flag is set to **Yes** by default.
12. **DD FREQUENCY** - Refers to the frequency with which detailed diagnosis measures are to be generated for this test. For a *Microsoft Hyper-V* server, this is set to *1:1* by default. This indicates that, by default, detailed measures will be generated every time this test runs, and also every time the test detects a problem. It is recommended that you do not change the default setting of this parameter. This is because, eG Enterprise can discover the IP addresses of the guest operating systems on a Hyper-V host, only while generating the detailed measures for this test. The automatic discovery of the guest IPs, in turn, enables eG Enterprise to perform **AutoVirtualMapping**.
13. **DETAILED DIAGNOSIS** - To make diagnosis more efficient and accurate, the eG suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option.

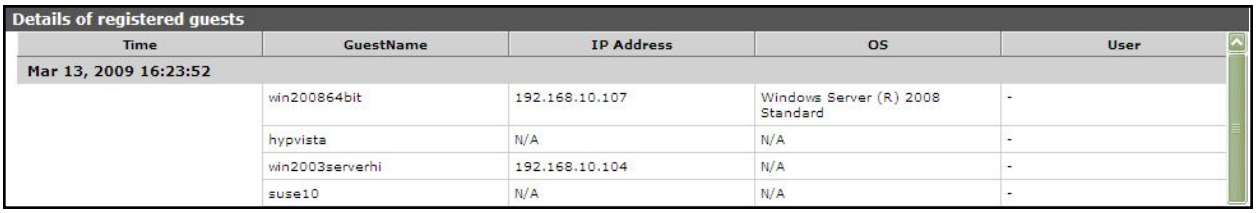
    The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:

    - The eG manager license should allow the detailed diagnosis capability
    - Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
Registered VMs	Indicates the total number of virtual machines that have been registered with the server currently.	Number	
VMs powered on	Indicates the number of guests that are currently powered on.	Number	To know which are the guests that are powered on, use the detailed diagnosis capability of this measure (if enabled).
VMs with users	Indicates the number of powered on guests with users logged in currently.	Number	To know which guests the users have logged into, use the detailed diagnosis capability of this measure (if enabled).
VMs without users	Indicates the number of powered on guests without any users logged in currently.	Number	To know which guests no user has logged into, use the detailed diagnosis capability of this measure (if enabled).
Added VMs	Indicates the number of guests that were newly added to the server during this measurement period.	Number	The detailed diagnosis of these measures, if enabled, lists the virtual machines that were migrated to or from (as the case may be) the Hyper-V server.
Removed VMs	Indicates the number of guests that were newly removed from the server during this measurement period.	Number	

The detailed diagnosis of the *Registered VMs* measure reports the name of the guests registered with the Hype-V server, the IP address of the guests, the guest OS, and the name of the user currently logged into the guest.

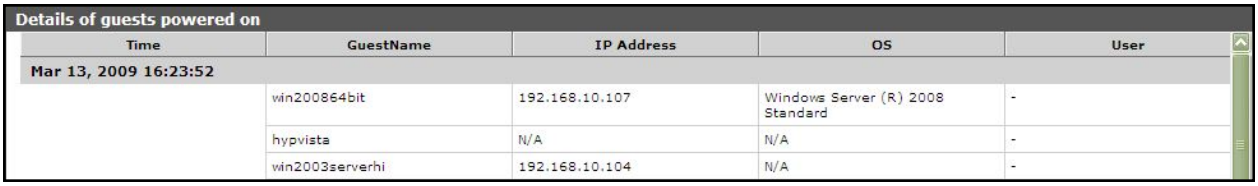


Details of registered guests

Time	GuestName	IP Address	OS	User
Mar 13, 2009 16:23:52				
	win200864bit	192.168.10.107	Windows Server (R) 2008 Standard	-
	hypvista	N/A	N/A	-
	win2003serverhi	192.168.10.104	N/A	-
	suse10	N/A	N/A	-

Figure 4.4: The detailed diagnosis of the Registered guests measure

The detailed diagnosis of the*VMs powered on* measure reports the name of the guests currently powered on, the IP address of the guests, the guest OS, and the name of the user currently logged into the guest.



Details of guests powered on

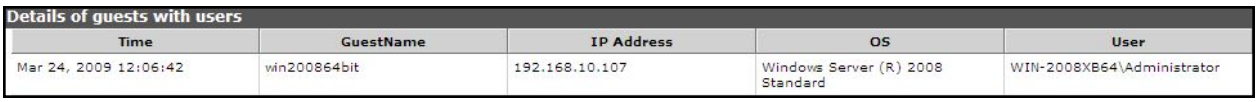
Time	GuestName	IP Address	OS	User
Mar 13, 2009 16:23:52				
	win200864bit	192.168.10.107	Windows Server (R) 2008 Standard	-
	hypvista	N/A	N/A	-
	win2003serverhi	192.168.10.104	N/A	-

Figure 4.5: The detailed diagnosis of the Guests powered on measure

Note:

The eG agent can extract the name and "outside view" metrics of Linux guests, but can neither discover the IP address nor report "inside view" metrics pertaining to Linux guests. Similarly, the eG agent cannot discover the IP address or obtain the "inside view" of those Windows VMs which do not support **Key/Value Pair Exchange** script

The detailed diagnosis of the *VMs with users* measure reveals the name, IP, and OS of the guests to which users are currently logged in, and the names of the users who have logged in.



Details of guests with users

Time	GuestName	IP Address	OS	User
Mar 24, 2009 12:06:42	win200864bit	192.168.10.107	Windows Server (R) 2008 Standard	WIN-2008XB64\Administrator

Figure 4.6: The detailed diagnosis of the VMs with users measure

4.1.3 Hyper-V Logins Test

This test monitors the user logins to guests and reports the total count of logins and logouts.

Target of the test:

A Hyper-V server hosting virtual desktops

Agent executing the test:

An internal agent

Output of the test

One set of results for the Hyper-V server monitored

Parameters of the test:

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured.
3. **PORT** - The port at which the **HOST** listens. By default, this is NULL.
4. **INSIDE VIEW USING** - By default, this test communicates with every VM remotely and extracts "inside view" metrics. Therefore, by default, the **INSIDE VIEW USING** flag is set to **Remote connection to VM (Windows)**.

   Typically, to establish this remote connection, eG Enterprise requires that the eG agent be configured with domain administrator privileges. In high-security environments, where the IT staff might have reservations about exposing the credentials of their domain administrators, this approach to extracting "inside view" metrics might not be preferred. In such environments therefore, eG Enterprise provides administrators the option to deploy a piece of software called the **eG VM Agent** on every Windows VM; this VM agent allows the eG agent to collect "inside view" metrics from the Windows VMs **without domain administrator rights.** Refer to Section 1.3.2 for more details on the **eG VM Agent**. To ensure that the "inside view" of Windows VMs is obtained using the eG VM Agent, set the **INSIDE VIEW USING** flag to **eG VM Agent (Windows)**. Once this is done, you can set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.
5. **DOMAIN, ADMIN USER, ADMIN PASSWORD,** and **CONFIRM PASSWORD** – By default, this test connects to each virtual guest remotely and attempts to collect "inside view" metrics. Accordingly, the **INSIDE VIEW USING** flag is set to **Remote connection to VM (Windows)** by default. To obtain a remote connection, the test must be configured with the privileges of an administrative user to the domain within which the guests reside. The first step towards this is to specify the **DOMAIN** within which the virtual guests reside. The **ADMIN USER** and **ADMIN PASSWORD** will change according to the **DOMAIN** specification. Discussed below are the different values that the **DOMAIN** parameter can take, and how they impact the **ADMIN USER** and **ADMIN PASSWORD** specifications:

   - **If the VMs belong to a single domain:** If the guests belong to a specific domain, then specify the name of that domain against the domain parameter. In this case, any administrative user in that domain will have remote access to all the virtual guests. Therefore, an administrator account in the given domain can be provided in the **ADMIN USER** field and the corresponding password in the **ADMIN PASSWORD** field. Confirm the password by retyping it in the **CONFIRM PASSWORD** text box.

- **If the VMs belong to different domains:**In this case, you might want to provide multiple domain names. If this is done, then, to access the guests in every configured domain, the test should be configured with the required user privileges; this implies that along with multiple **DOMAIN** names, multiple **ADMIN USER** names and **ADMIN PASSWORDS** would also have to be provided. To help administrators provide these user details quickly and easily, the eG administrative interface embeds a special configuration page. To access this page, simply click on the **Click here** hyperlink that appears just above the parameters of this test in the test configuration page. To know how to use the special page, refer to Section 4.1.3 of this document.
- **If the INSIDE VIEW USING flag is set to 'eG VM Agent (Windows)':** On the other hand, if the **INSIDE VIEW USING** flag is set to **eG VM Agent (Windows)**, then it implies that the inside view can be obtained without domain administrator privileges. Therefore, set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.

6. **REPORT BY USER** – For the *Microsoft Hyper-V* monitoring model, the **REPORT BY USER** flag is set to **NO** by default, indicating that by default, the guest operating systems on the Hyper-V server are identified using the hostname specified in the operating system. On the other hand, for the *Microsoft Hyper-V VDI* model, this flag is set to **YES** by default; this implies that in case of VDI servers, by default, the guests will be identified using the login of the user who is accessing the guest OS. In other words, in VDI environments, this test will, by default, report measures for every *username_on_virtualmachinename*.
7. **DD FREQUENCY** - Refers to the frequency with which detailed diagnosis measures are to be generated for this test. For a *Microsoft Hyper-V* server, this is set to *1:1* by default. This indicates that, by default, detailed measures will be generated every time this test runs, and also every time the test detects a problem. It is recommended that you do not change the default setting of this parameter. This is because, eG Enterprise can discover the IP addresses of the guest operating systems on a Hyper-V host, only while generating the detailed measures for this test. The automatic discovery of the guest IPs, in turn, enables eG Enterprise to perform **AutoVirtualMapping**.
8. **DETAILED DIAGNOSIS** - To make diagnosis more efficient and accurate, the eG suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option.

   The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:

   - The eG manager license should allow the detailed diagnosis capability
   - Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measurement	Description	Measurement Unit	Interpretation
Current sessions	Indicates the number of user sessions that are currently active across all guests	Number	This is a good indicator of the session load on the guests.
New logins	Indicates the number of new logins to the guests.	Number	If this measure reports a non-zero value, use the detailed diagnosis of the measure to know which user logged into which VM, when.
Percentage of new logins	Indicates the percentage of current sessions that logged in during the last measurement period.	Percent	
Sessions logging out	Indicates the number of sess ions that logged out.	Number	If all the current sessions suddenly log out, it indicates a problem condition that requires investigation. The detailed diagnosis of this measure lists the sessions that logged out.

4.1.4 VDI Applications Test

This test discovers the applications executing on the virtual desktops and reports the availability and resource-usage of each of the desktop applications.

Target of the test:

A Hyper-V / Hyper-V VDI server

Agent executing the test:

An internal agent

Output of the test

One set of results each distinct application that is being accessed by users of virtual desktops

Parameters of the test:

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured.
3. **PORT** - The port at which the **HOST** listens. By default, this is NULL.
4. **INSIDE VIEW USING** - By default, this test communicates with every VM remotely and extracts "inside view" metrics. Therefore, by default, the **INSIDE VIEW USING** flag is set to **Remote connection to VM (Windows)**.

   Typically, to establish this remote connection, eG Enterprise requires that the eG agent be configured with domain administrator privileges. In high-security environments, where the IT staff might have reservations about exposing the credentials of their domain administrators, this approach to extracting "inside view" metrics might not be preferred. In such environments therefore, eG Enterprise provides administrators the option to deploy a piece of software called the **eG VM Agent** on every Windows VM; this VM agent allows the eG agent to collect "inside view" metrics from the Windows VMs **without domain administrator rights.** Refer to Section 1.3.2 for more details on the **eG VM Agent**. To ensure that the "inside view" of Windows VMs is obtained using the eG VM Agent, set the **INSIDE VIEW USING** flag to **eG VM Agent (Windows)**. Once this is done, you can set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.
5. **DOMAIN, ADMIN USER, ADMIN PASSWORD,** and **CONFIRM PASSWORD** – By default, this test connects to each virtual guest remotely and attempts to collect "inside view" metrics. Accordingly, the **INSIDE VIEW USING** flag is set to **Remote connection to VM (Windows)** by default. To obtain a remote connection, the test must be configured with the privileges of an administrative user to the domain within which the guests reside. The first step towards this is to specify the **DOMAIN** within which the virtual guests reside. The **ADMIN USER** and **ADMIN PASSWORD** will change according to the **DOMAIN** specification. Discussed below are the different values that the **DOMAIN** parameter can take, and how they impact the **ADMIN USER** and **ADMIN PASSWORD** specifications:

   - **If the VMs belong to a single domain:** If the guests belong to a specific domain, then specify the name of that domain against the domain parameter. In this case, any administrative user in that domain will have remote access to all the virtual guests. Therefore, an administrator account in the given domain can be provided in the **ADMIN USER** field and the corresponding password in the **ADMIN PASSWORD** field. Confirm the password by retyping it in the **CONFIRM PASSWORD** text box.
   - **If the VMs belong to different domains:**In this case, you might want to provide multiple domain names. If this is done, then, to access the guests in every configured domain, the test should be configured with the required user privileges; this implies that along with multiple **DOMAIN** names, multiple **ADMIN USER** names and **ADMIN PASSWORDS** would also have to be provided. To help administrators provide these user details quickly and easily, the eG administrative interface embeds

a special configuration page. To access this page, simply click on the **Click here** hyperlink that appears just above the parameters of this test in the test configuration page. To know how to use the special page, refer to Section 4.1.4 of this document.

- **If the INSIDE VIEW USING flag is set to 'eG VM Agent (Windows)':** On the other hand, if the **INSIDE VIEW USING** flag is set to **eG VM Agent (Windows)**, then it implies that the inside view can be obtained without domain administrator privileges. Therefore, set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.

6. **REPORT BY USER** – For the *Microsoft Hyper-V* monitoring model, the **REPORT BY USER** flag is set to **NO** by default, indicating that by default, the guest operating systems on the Hyper-V server are identified using the hostname specified in the operating system. On the other hand, for the *Microsoft Hyper-V VDI* model, this flag is set to **YES** by default; this implies that in case of VDI servers, by default, the guests will be identified using the login of the user who is accessing the guest OS. In other words, in VDI environments, this test will, by default, report measures for every *username_on_virtualmachinename*.
7. **REPORT POWERED OS** - **This flag becomes relevant only if the REPORT BY USER flag is set to 'Yes'.**

   If the **REPORT POWERED OS** flag is set to **Yes** (which is the default setting), then this test will report measures for even those VMs that do not have any users logged in currently. Such guests will be identified by their *virtualmachine name* and not by the *username_on_virtualmachinename*. On the other hand, if the **REPORT POWERED OS** flag is set to **No**, then this test will not report measures for those VMs to which no users are logged in currently.
8. **IGNORE VMS INSIDE VIEW** - Administrators of some high security Hyper-V environments might not have permissions to internally monitor one/more VMs. The eG agent can be configured to **not obtain the 'inside view' of such 'inaccessible' VMs** using the **IGNORE VMS INSIDE VIEW** parameter. Against this parameter, you can provide a comma-separated list of VM names, or VM name patterns, for which the inside view need not be obtained. For instance, your **IGNORE VMS INSIDE VIEW** specification can be: **xp,*lin*,win*,vista*. Here, the * (asterisk) is used to denote leading and trailing spaces (as the case may be). By default, this parameter is set to *none* indicating that the eG agent obtains the inside view of all VMs on a Hyper-V host by default.

   Note:

   While performing VM discovery, the eG agent will not discover the operating system of the VMs configured in the **IGNORE VMS INSIDE VIEW** text box.

9. **EXCLUDE VMS** - Administrators of some virtualized environments may not want to monitor some of their less-critical VMs - for instance, VM templates - both from 'outside' and from 'inside'. The eG agent in this case can be configured to completely exclude such VMs from its monitoring purview. To achieve this, provide a comma-separated list of VMs to be excluded from monitoring in the **EXCLUDE VMS** text box. Instead of VMs, VM name patterns can also be provided here in a comma-separated list. For

example, your **EXCLUDE VMS** specification can be: **xp,*lin*,win*,vista*. Here, the * (asterisk) is used to denote leading and trailing spaces (as the case may be). By default, this parameter is set to *none* indicating that the eG agent obtains the inside and outside views of all VMs on a virtual host by default. By providing a comma-separated list of VMs/VM name patterns in the **EXCLUDE VMS** text box, you can make sure the eG agent stops collecting 'inside' and 'outside' view metrics for a configured set of VMs.

10. **IGNORE WINNT** – By default, the eG agent does not support the *inside view* for VMs executing on **Windows NT** operating systems. Accordingly, the **IGNORE WINNT** flag is set to **Yes** by default.
11. **IS SHOW ALL APPS** - To ensure that the test monitors only specific applications executing on the desktops and not all of them, set the **IS SHOW ALL APPS** flag to **No**. Once this is done, then, you need to configure those applications that you want to exclude from the monitoring scope of this test. For this purpose, follow the steps given below:
    - Edit the eg_tests.ini file (in the {EG_INSTALL_DIR}\manager\config directory).
    - In the [EXCLUDE_APPLICATIONS] section of the file, you will find an entry of the following format:

      VmgApplicationTest={Comma-separated list of applications to be excluded}
    - To the comma-separated application list that pre-exists, append the applications that you want to monitor. For instance, if your test need not monitor notepad.exe, and powerpnt.exe, then, your entry should be:

      VmgApplicationTest=.....,notepad.exe,powerpnt.exe

      Note that the exact application names should be provided, but the extensions (for instance, .exe) can be dispensed with.
12. **SHOW USER APPS ONLY** - By default, this flag is set to **Yes**. Accordingly, this test will monitor only those applications/processes that are running in the user's account. To monitor all applications/processes running in the virtual desktops, regardless of the user account using which they are running, set this flag to **No**.
13. **DD FREQUENCY** - Refers to the frequency with which detailed diagnosis measures are to be generated for this test. For a *Microsoft Hyper-V* server, this is set to *1:1* by default. This indicates that, by default, detailed measures will be generated every time this test runs, and also every time the test detects a problem. It is recommended that you do not change the default setting of this parameter. This is because, eG Enterprise can discover the IP addresses of the guest operating systems on a Hyper-V host, only while generating the detailed measures for this test. The automatic discovery of the guest IPs, in turn, enables eG Enterprise to perform **AutoVirtualMapping**.
14. **DETAILED DIAGNOSIS** - To make diagnosis more efficient and accurate, the eG suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option.

The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:

- The eG manager license should allow the detailed diagnosis capability
- Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

\

Measurement	Description	Measurement Unit	Interpretation
Processes running	Indicates the number of instances of this application that is currently executing across all virtual desktops on the target host operating system.	Number	
CPU usage	Indicates the percentage of physical CPU resources utilized	Percent	A very high value of this measure is a cause for concern,

Measurement	Description	Measurement Unit	Interpretation
	by this application across the guest VMs.		as it indicates excessive CPU usage by a single application. This in turn would cause other desktop applications to contend for limited physical resources, thus degrading the performance of those applications and that of the virtual server as a whole.
Memory usage	Indicates the percentage of physical memory resources utilized by this application across the guest VMs.	Percent	A very high value of this measure is a cause for concern, as it indicates excessive memory usage by a single application. This in turn would cause other desktop applications to contend for limited physical memory resources, thus degrading the performance of those applications and that of the virtual server as a whole.
CPU used	Indicates the physcial CPU (in Mhz) used up by this application.	Mhz	

4.2 The Inside View of Desktops Layer

The **Outside View of VMs layer** provides an “external” view of the different VM guests – the metrics reported at this layer are based on what the VMware host is seeing about the performance of the individual guests. However, an external view of the VM guest operating system and its applications may not be sufficient. For instance, suppose one of the disk partitions of the guest operating system has reached capacity. This information cannot be gleaned from host operating system. Likewise, bottlenecks such as a longer process run queue or a higher disk queue length are more visible using an internal monitor. Internal monitoring (from within the guest operating system) also provides details about the resource utilization of different application (s) or processes.

The tests mapped to the **Inside View of Desktops** layer provide an "internal" view of the workings of each of the guests - these tests execute on an Hyper-V host, but send probes into each of the guest operating systems to analyze how well each guest utilizes the resources that are allocated to it, and how well it handles user sessions, TCP traffic, and network loading.

By default however, clicking on the **Inside View of Desktops** layer, does not list the associated tests. Instead, Figure 4.7 appears. This figure displays the current state of all virtual desktops that have been configured on the monitored Hyper-V host.

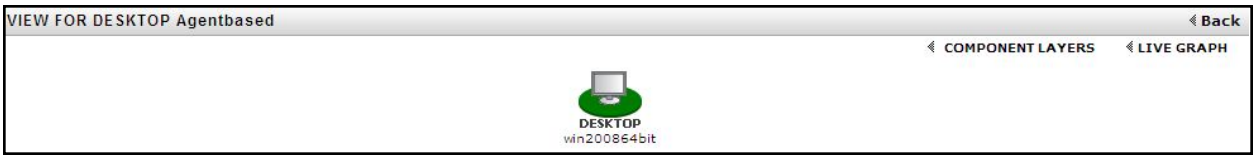


Figure 4.7: The current state of the desktops configured on the Hyper-V host that is monitored

Clicking on any of the guests in the **Desktop view** leads you to Figure 4.8 that displays all the performance metrics extracted from that guest, in real-time. You are thus enabled to cross-correlate across the various metrics, and quickly detect the root-cause of current/probable disturbances to the internal health of a guest. To view the time-of-day variations in a measure, you can view its graph by clicking on that measure in Figure 4.8.

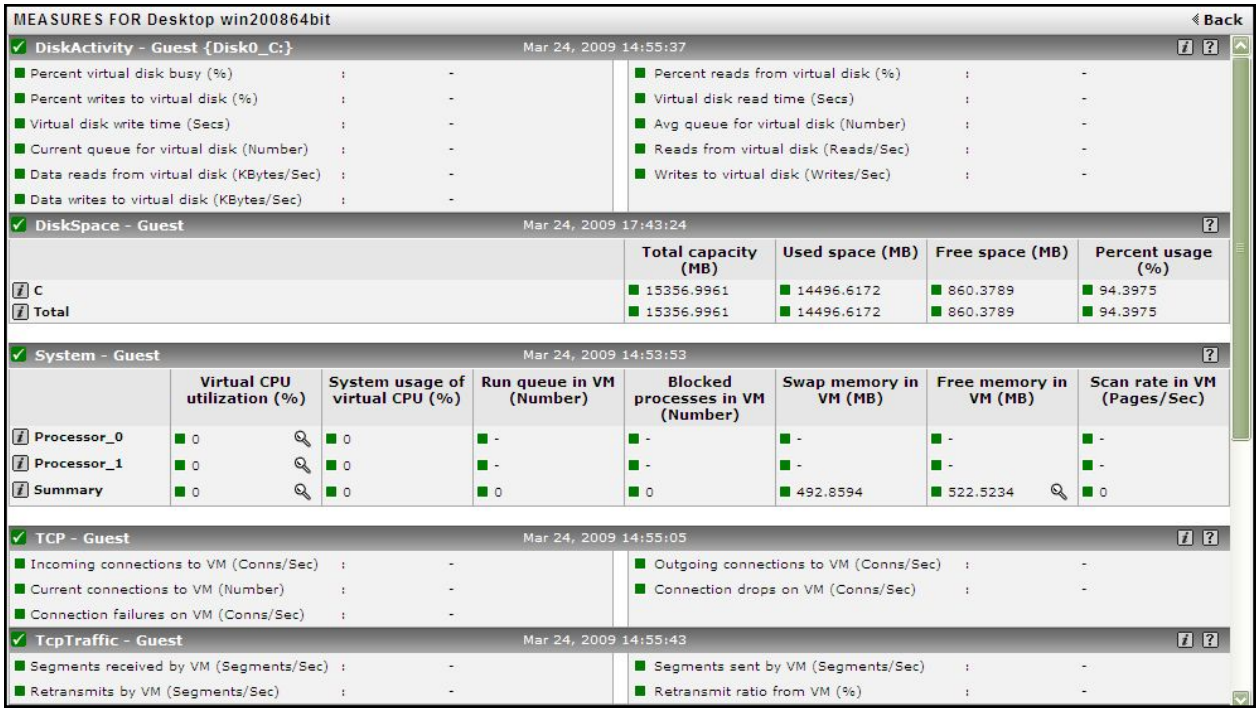


Figure 4.8: Measures pertaining to a chosen guest

To view real-time graphs of pre-configured measures (pertaining to the *Microsoft Hyper-V* host and the guests operating on it), click on the **LIVE GRAPH** link in Figure 4.7. Alternatively, you can click on the icon that appears in the **Tests** panel (see Figure 4.2) when the **Outside View of VMs layer** is clicked. The graph display that appears subsequently (see Figure 4.9) has been organized in such a way that next to every host-

pertinent measure graph, the closely related guest-specific measure graph appears. For instance, next to the graph of the 'Cpu utilization' measure of the *Hyper-V Logical Processors* test, you will find a graph of the 'Virtual machine cpu utilization' measure of the *Hyper-V VMs* test. This way, you can easily compare and correlate how well the physical CPU resources are being utilized by both the Hyper-V host and the guests. On the basis of this analysis, you can proactively isolate potential performance issues, and also determine the root-cause of the issue - is it the Hyper-V host? or is it the virtual guest? If you access this page from the **LIVE GRAPH** link in Figure 4.7, then, by default, you will view live graphs pertaining to the *Hyper-V VDI* server. However, you can select a different virtualized component-type and a different virtualized component using the **type** and **ComponentName** lists (respectively) in Figure 4.7.

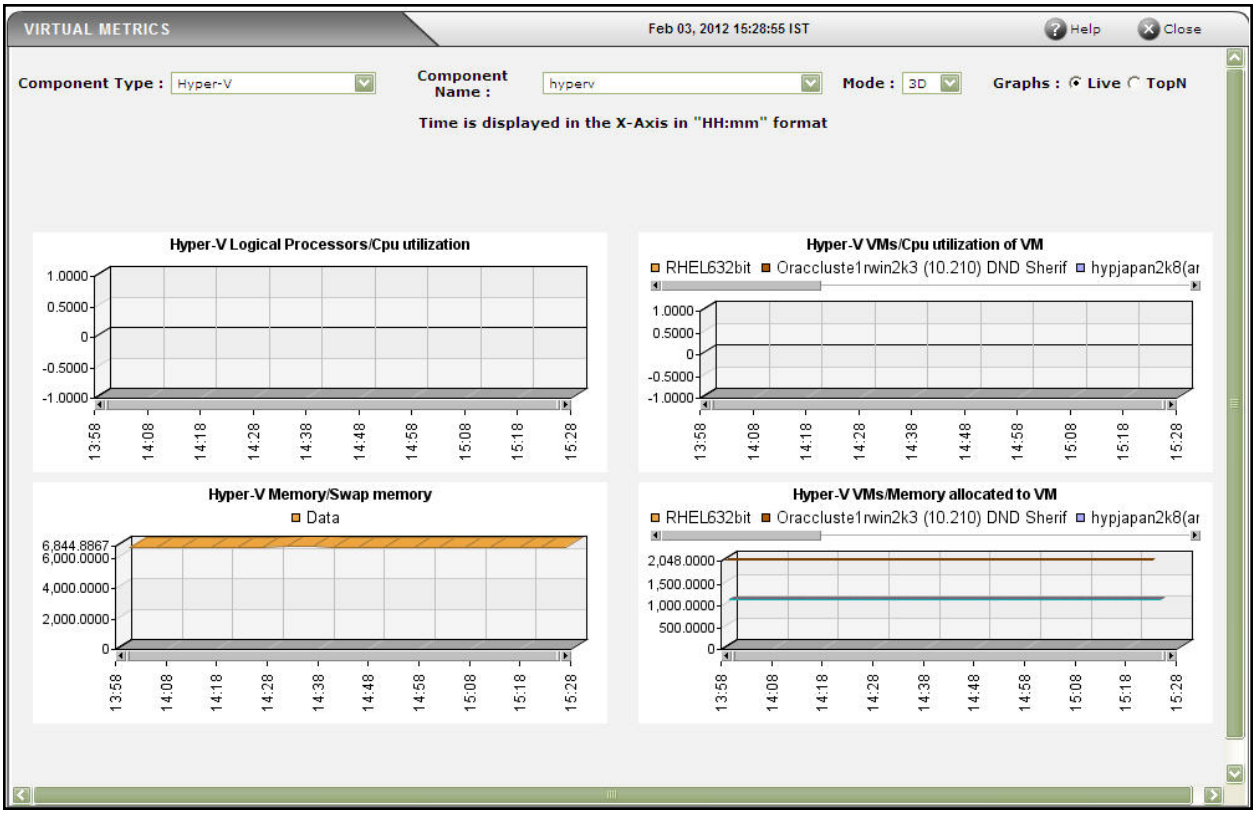


Figure 4.9: Live graph comparing physical resource usage of a Hyper-V VDI server (on the left) and resource usage levels of the individual VMs (on the right)

To return to the layer model of the *Microsoft Hyper-V VDI* server and view the tests mapped to the **Inside View of Desktops** layer, click on the **COMPONENT LAYERS** link in Figure 4.7. The tests depicted by Figure 4.10 then appear.

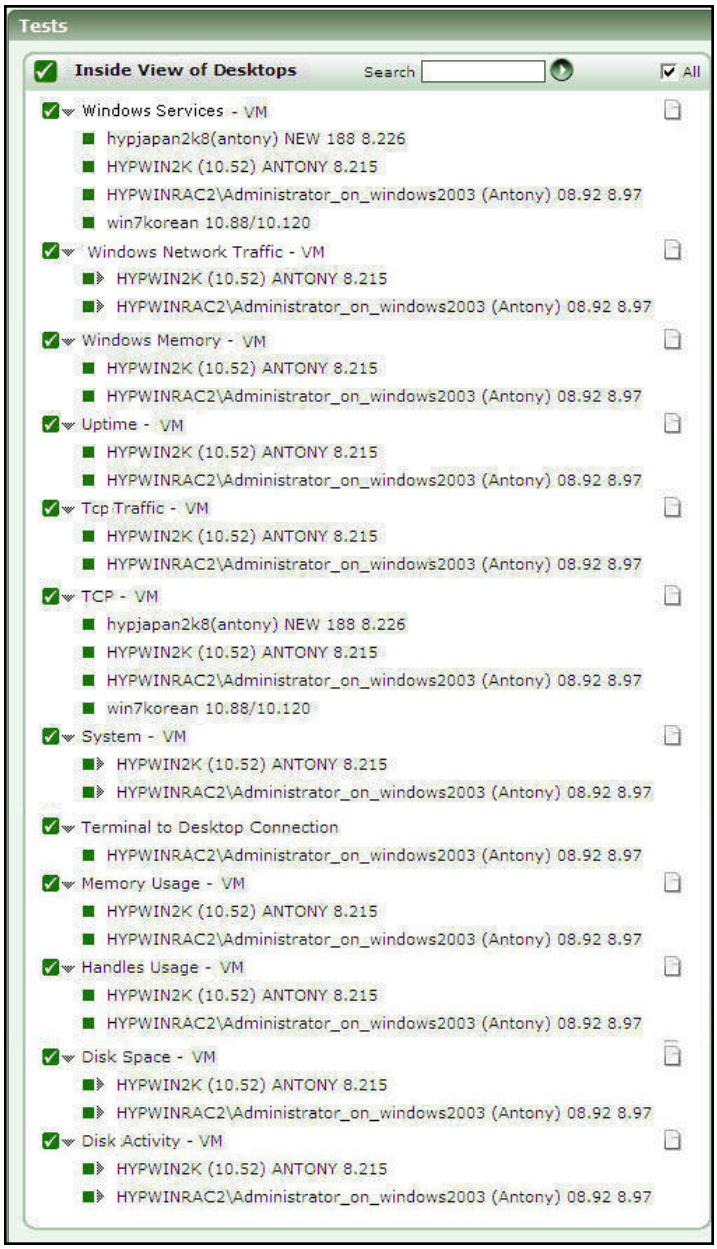


Figure 4.10: The tests associated with the Inside View of Desktops layer

Alternatively, you can also configure eG to first display the tests mapped to the **Inside View of Desktops** layer first upon clicking it, and not the **Desktop View**. For this, follow the steps given below:

- Edit the **eg_ui.ini** file in the **<EG_INSTALL_DIR>\manager\config** directory
- Set the **LAYERMODEL_LINK_TO_VIRTUAL** flag in the file to **false**; this is set to **true** by default.
- Save the **eg_ui.ini** file.

4.2.1 Virtual Desktop Client's Network Connection Test

A Virtual Desktop Infrastructure (VDI) is a shared environment in which multiple users connect to desktops hosted by virtual machines executing on a Hyper-V host from remote terminals. One of the key factors influencing user experience in such an environment is the latency seen by the users when connecting to a virtual desktop. High network latencies or packet losses during transmission can cause significant slow-downs in request processing by the desktop. Hence, monitoring latencies between the virtual desktop and individual client terminals is important.

The Virtual Desktop Client's Network Connection test is executed by the eG agent on a Hyper-V host. This test auto-discovers the virtual desktops on the host, the users who are currently logged on to each of the virtual desktops, and the IP address from which they are connecting to the virtual desktops. For each user, the test monitors the quality of the link between the client and the virtual desktop.

Using this test, an administrator can identify user sessions that are being impacted by high latencies or by excessive packet drops. In some cases, a virtual desktop may regard a user session as active, even though the network link connecting the user terminal to the virtual desktop has failed. The Terminal to Desktop Connection test alerts administrators to such situations.

This test will work on Windows VMs only.

Target of the test:

A Hyper-V server hosting virtual desktops

Agent executing the test:

An internal agent

Output of the test:

One set of results will be reported for the user who is currently logged into each Windows virtual desktop on the server

Parameters of the test:

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured.
3. **PORT** - The port at which the **HOST** listens. By default, this is NULL.

4. **INSIDE VIEW USING** - By default, this test communicates with every VM remotely and extracts "inside view" metrics. Therefore, by default, the **INSIDE VIEW USING** flag is set to **Remote connection to VM (Windows)**.

   Typically, to establish this remote connection, eG Enterprise requires that the eG agent be configured with domain administrator privileges. In high-security environments, where the IT staff might have reservations about exposing the credentials of their domain administrators, this approach to extracting "inside view" metrics might not be preferred. In such environments therefore, eG Enterprise provides administrators the option to deploy a piece of software called the **eG VM Agent** on every Windows VM; this VM agent allows the eG agent to collect "inside view" metrics from the Windows VMs **without domain administrator rights.** Refer to Section 1.3.2 for more details on the **eG VM Agent**. To ensure that the "inside view" of Windows VMs is obtained using the eG VM Agent, set the **INSIDE VIEW USING** flag to **eG VM Agent (Windows)**. Once this is done, you can set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.

5. **DOMAIN, ADMIN USER, ADMIN PASSWORD,** and **CONFIRM PASSWORD** – By default, this test connects to each virtual guest remotely and attempts to collect "inside view" metrics. Accordingly, the **INSIDE VIEW USING** flag is set to **Remote connection to VM (Windows)** by default. To obtain a remote connection, the test must be configured with the privileges of an administrative user to the domain within which the guests reside. The first step towards this is to specify the **DOMAIN** within which the virtual guests reside. The **ADMIN USER** and **ADMIN PASSWORD** will change according to the **DOMAIN** specification. Discussed below are the different values that the **DOMAIN** parameter can take, and how they impact the **ADMIN USER** and **ADMIN PASSWORD** specifications:

   - **If the VMs belong to a single domain:** If the guests belong to a specific domain, then specify the name of that domain against the domain parameter. In this case, any administrative user in that domain will have remote access to all the virtual guests. Therefore, an administrator account in the given domain can be provided in the **ADMIN USER** field and the corresponding password in the **ADMIN PASSWORD** field. Confirm the password by retyping it in the **CONFIRM PASSWORD** text box.
   - **If the VMs belong to different domains:**In this case, you might want to provide multiple domain names. If this is done, then, to access the guests in every configured domain, the test should be configured with the required user privileges; this implies that along with multiple **DOMAIN** names, multiple **ADMIN USER** names and **ADMIN PASSWORDS** would also have to be provided. To help administrators provide these user details quickly and easily, the eG administrative interface embeds a special configuration page. To access this page, simply click on the encircled '+' icon that appears alongside the . To know how to use the special page, refer to Section 3.6.1 of this document.
   - **If the INSIDE VIEW USING flag is set to 'eG VM Agent (Windows)'**: On the other hand, if the **INSIDE VIEW USING** flag is set to **eG VM Agent (Windows)**, then it implies that the inside view can be obtained without domain administrator privileges. Therefore, set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.

4. **REPORT BY USER** – For the *Hyper-V* monitoring model, the **REPORT BY USER** flag is set to **NO** by default, indicating that by default, the guest operating systems on the Hyper-V server are identified using the hostname specified in the operating system. On the other hand, for the *Hyper-V VDI* model, this flag is set to **YES** by default; this implies that in case of VDI servers, by default, the guests will be

identified using the login of the user who is accessing the guest OS. In other words, in VDI environments, this test will, by default, report measures for every *username_on_virtualmachinename*.

5. **REPORT POWERED OS** - **This flag becomes relevant only if the REPORT BY USER flag is set to 'Yes'.**

   If the **REPORT POWERED OS** flag is set to **Yes** (which is the default setting), then this test will report measures for even those VMs that do not have any users logged in currently. Such guests will be identified by their *virtualmachine name* and not by the *username_on_virtualmachinename*. On the other hand, if the **REPORT POWERED OS** flag is set to **No**, then this test will not report measures for those VMs to which no users are logged in currently.

6. **IGNORE VMS INSIDE VIEW** - Administrators of some high security Hyper-V environments might not have permissions to internally monitor one/more VMs. The eG agent can be configured to **not obtain the 'inside view' of such 'inaccessible' VMs** using the **IGNORE VMS INSIDE VIEW** parameter. Against this parameter, you can provide a comma-separated list of VM names, or VM name patterns, for which the inside view need not be obtained. For instance, your **IGNORE VMS INSIDE VIEW** specification can be: **xp,*lin*,win*,vista*. Here, the * (asterisk) is used to denote leading and trailing spaces (as the case may be). By default, this parameter is set to *none* indicating that the eG agent obtains the inside view of all VMs on a Hyper-V host by default.

   Note:
   While performing VM discovery, the eG agent will not discover the operating system of the VMs configured in the **IGNORE VMS INSIDE VIEW** text box.

7. **EXCLUDE VMS** - Administrators of some virtualized environments may not want to monitor some of their less-critical VMs - for instance, VM templates - both from 'outside' and from 'inside'. The eG agent in this case can be configured to completely exclude such VMs from its monitoring purview. To achieve this, provide a comma-separated list of VMs to be excluded from monitoring in the **EXCLUDE VMS** text box. Instead of VMs, VM name patterns can also be provided here in a comma-separated list. For example, your **EXCLUDE VMS** specification can be: **xp,*lin*,win*,vista*. Here, the * (asterisk) is used to denote leading and trailing spaces (as the case may be). By default, this parameter is set to *none* indicating that the eG agent obtains the inside and outside views of all VMs on a virtual host by default. By providing a comma-separated list of VMs/VM name patterns in the **EXCLUDE VMS** text box, you can make sure the eG agent stops collecting 'inside' and 'outside' view metrics for a configured set of VMs.

8. **IGNORE WINNT** – By default, the eG agent does not support the *inside view* for VMs executing on **Windows NT** operating systems. Accordingly, the **IGNORE WINNT** flag is set to **Yes** by default.

9. **PACKETSIZE** - The size of packets used for the test (in bytes)

10. **PACKETCOUNT** - The number of packets exchanged between the virtual desktop and the user terminal during the test

11. **TIMEOUT** - How long after transmission should a packet be deemed lost (in seconds)

12. **PACKETINTERVAL** - Represents the interval (in milliseconds) between successive packet transmissions during the execution of this test.

13. **REPORTUNAVAILABILITY** – By default, this flag is set to **No**. This implies that, by default, the test will

not report the unavailability of network connection between a user terminal and a virtual desktop. In other words, if the *Packet loss* measure of this test registers the value 100% for any user, then, by default, this test will not report any measure for that user; under such circumstances, the corresponding user name will not appear as a descriptor of this test. You can set this flag to **Yes**, if you want the test to report and alert you to the unavailability of network connection between a user terminal and a virtual desktop.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
Number of sessions	Indicates the current number of sessions for a particular user	Number	The value 0 indicates that the user is not currently connected to the virtual desktop.
Average delay	Indicates the average delay between transmission of a request by the agent on a virtual desktop and receipt of the response back from the user terminal.	Secs	Comparing the value of this measure across users will enable administrators to quickly and accurately identify users who are experiencing higher latency when connecting to a virtual desktop.
Minimum delay	Indicates the minimum delay between transmission of a	Secs	A significant increase in the minimum round-trip time is often a sure sign of a poor link between

Measurement	Description	Measurement Unit	Interpretation
	request by the agent on a virtual desktop and receipt of the response back from the user terminal.		the desktop and a user's terminal.
Packet loss	Indicates the percentage of packets lost during data exchange between the virtual desktop and the user terminal.	Percent	Comparing the value of this measure across users will enable administrators to quickly and accurately identify users who are experiencing slowdowns because of poor performance on the network links between their terminals and the virtual desktop.

Note:

- If the same user is connecting to the virtual desktop from multiple client terminals, the value of the *Number of sessions*, *Average delay*, and *Packet loss* measures will be averaged across all the sessions of that user. The *Minimum delay* measure, on the other hand, will display the least value reported for *Minimum delay* across all the sessions of that user.
- When a user logs out, the number of sessions will be reduced by 1. If the number of user sessions becomes 0, the corresponding entry for that user in the eG user interface will be removed after a short period of time.

4.2.2 Desktop's HDX Channel Test

As already mentioned, the key factors influencing user experience in a virtual desktop infrastructure are the latencies experienced by the user while connecting to the desktop via ICA and the bandwidth used when a user interacts with a virtual desktop. High latency and excessive bandwidth consumption can often slowdown access to desktops, thereby significantly delaying subsequent user operations. Hence, monitoring the latency and bandwidth usage of the ICA communication channel between the user terminal and the virtual desktops is essential.

The Desktop's HDX Channel test auto-discovers the virtual desktops on the Hyper-V host and the users who are currently connected to each desktop. For each such user, the test monitors the communication between a user and the virtual desktop, and reports the following:

The latency experienced by each user session;

The bandwidth used by the incoming and outgoing data/audio/multimedia traffic transacted by the ICA communication channel between each user and virtual desktop;

Using this test, an administrator can identify user sessions that are being impacted by high latency and abnormal bandwidth usage. In addition, the test also reveals the type of traffic that is causing excessive bandwidth usage, thereby providing pointers to how the client configuration can be fine-tuned in order to reduce bandwidth consumption and improve performance.

This test is disabled by default. To enable the test, go to the **ENABLE / DISABLE TESTS** page using the menu sequence: Agents -> Tests -> Enable/Disable, pick *Microsoft Hyper-V - VDI* as the **Component type**, set *Performance* as the **Test type**, choose this test from the **DISABLED TESTS** list, and click on the >> button to move the test to the **ENABLED TESTS** list.

Note:

This test will report metrics only if the following conditions are fulfilled:

- The test is applicable to Windows VMs only.
- The VMs being monitored should be managed by XenDesktop Broker.
- The Virtual Desktop Agent software should have been installed on the VMs.
- The **ICA Session** performance object should be enabled on the VMs.

Target of the test:

A Hyper-V server hosting virtual desktops

Agent executing the test:

An internal agent

Output of the test:

One set of results will be reported for every user who is connected to a virtual desktop, via ICA

Parameters of the test:

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured.
3. **PORT** - The port at which the **HOST** listens. By default, this is NULL.
4. **INSIDE VIEW USING** - By default, this test communicates with every VM remotely and extracts "inside view" metrics. Therefore, by default, the **INSIDE VIEW USING** flag is set to **Remote connection to VM (Windows)**.

   Typically, to establish this remote connection, eG Enterprise requires that the eG agent be configured with domain administrator privileges. In high-security environments, where the IT staff might have reservations about exposing the credentials of their domain administrators, this approach to extracting "inside view" metrics might not be preferred. In such environments therefore, eG Enterprise provides administrators the option to deploy a piece of software called the **eG VM Agent** on every Windows VM; this VM agent allows the eG agent to collect "inside view" metrics from the Windows VMs **without domain administrator rights.** Refer to Section 1.3.2 for more details on the **eG VM Agent**. To ensure that the "inside view" of Windows VMs is obtained using the eG VM Agent, set the **INSIDE VIEW USING** flag to **eG VM Agent (Windows)**. Once this is done, you can set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.
5. **DOMAIN, ADMIN USER, ADMIN PASSWORD,** and **CONFIRM PASSWORD** – By default, this test connects to each virtual guest remotely and attempts to collect "inside view" metrics. Accordingly, the **INSIDE VIEW USING** flag is set to **Remote connection to VM (Windows)** by default. To obtain a remote connection, the test must be configured with the privileges of an administrative user to the domain within which the guests reside. The first step towards this is to specify the **DOMAIN** within which the virtual guests reside. The **ADMIN USER** and **ADMIN PASSWORD** will change according to the **DOMAIN** specification. Discussed below are the different values that the **DOMAIN** parameter can take, and how they impact the **ADMIN USER** and **ADMIN PASSWORD** specifications:
   - **If the VMs belong to a single domain:** If the guests belong to a specific domain, then specify the name of that domain against the domain parameter. In this case, any administrative user in that domain will have remote access to all the virtual guests. Therefore, an administrator account in the given domain can be provided in the **ADMIN USER** field and the corresponding password in the **ADMIN PASSWORD** field. Confirm the password by retyping it in the **CONFIRM PASSWORD** text box.
   - **If the VMs belong to different domains:**In this case, you might want to provide multiple domain names. If this is done, then, to access the guests in every configured domain, the test should be configured with the required user privileges; this implies that along with multiple **DOMAIN** names, multiple **ADMIN USER** names and **ADMIN PASSWORDS** would also have to be provided. To help administrators provide these user details quickly and easily, the eG administrative interface embeds a special configuration page. To access this page, simply click on the encircled '+' icon that appears

alongside the . To know how to use the special page, refer to Section 3.6.1 of this document.

- **If the INSIDE VIEW USING flag is set to 'eG VM Agent (Windows)'**: On the other hand, if the **INSIDE VIEW USING** flag is set to **eG VM Agent (Windows)**, then it implies that the inside view can be obtained without domain administrator privileges. Therefore, set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.

4. **REPORT BY USER** – For the *Hyper-V* monitoring model, the **REPORT BY USER** flag is set to **NO** by default, indicating that by default, the guest operating systems on the Hyper-V server are identified using the hostname specified in the operating system. On the other hand, for the *Hyper-V VDI* model, this flag is set to **YES** by default; this implies that in case of VDI servers, by default, the guests will be identified using the login of the user who is accessing the guest OS. In other words, in VDI environments, this test will, by default, report measures for every *username_on_virtualmachinename*.
5. **REPORT POWERED OS** - **This flag becomes relevant only if the REPORT BY USER flag is set to 'Yes'.**

   If the **REPORT POWERED OS** flag is set to **Yes** (which is the default setting), then this test will report measures for even those VMs that do not have any users logged in currently. Such guests will be identified by their *virtualmachine name* and not by the *username_on_virtualmachinename*. On the other hand, if the **REPORT POWERED OS** flag is set to **No**, then this test will not report measures for those VMs to which no users are logged in currently.
6. **IGNORE VMS INSIDE VIEW** - Administrators of some high security Hyper-V environments might not have permissions to internally monitor one/more VMs. The eG agent can be configured to **not obtain the 'inside view' of such 'inaccessible' VMs** using the **IGNORE VMS INSIDE VIEW** parameter. Against this parameter, you can provide a comma-separated list of VM names, or VM name patterns, for which the inside view need not be obtained. For instance, your **IGNORE VMS INSIDE VIEW** specification can be: **xp,*lin*,win*,vista*. Here, the * (asterisk) is used to denote leading and trailing spaces (as the case may be). By default, this parameter is set to *none* indicating that the eG agent obtains the inside view of all VMs on a Hyper-V host by default.

   Note:
   While performing VM discovery, the eG agent will not discover the operating system of the VMs configured in the **IGNORE VMS INSIDE VIEW** text box.
7. **EXCLUDE VMS** - Administrators of some virtualized environments may not want to monitor some of their less-critical VMs - for instance, VM templates - both from 'outside' and from 'inside'. The eG agent in this case can be configured to completely exclude such VMs from its monitoring purview. To achieve this, provide a comma-separated list of VMs to be excluded from monitoring in the **EXCLUDE VMS** text box. Instead of VMs, VM name patterns can also be provided here in a comma-separated list. For example, your **EXCLUDE VMS** specification can be: **xp,*lin*,win*,vista*. Here, the * (asterisk) is used to denote leading and trailing spaces (as the case may be). By default, this parameter is set to *none* indicating that the eG agent obtains the inside and outside views of all VMs on a virtual host by default. By providing a comma-separated list of VMs/VM name patterns in the **EXCLUDE VMS** text box, you can make sure the eG agent stops collecting 'inside' and 'outside' view metrics for a configured set of VMs.
8. **IGNORE WINNT** – By default, the eG agent does not support the *inside view* for VMs executing on **Windows NT** operating systems. Accordingly, the **IGNORE WINNT** flag is set to **Yes** by default.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
Session average latency	Indicates the average client latency over the lifetime of this session.	Secs	Comparing the value of this measure across users will enable administrators to quickly and accurately identify users who are experiencing higher latency when connecting to a virtual desktop.
Session deviation latency	Indicates the difference between the minimum and maximum measured latency values for this session.	Secs	
Audio bandwidth output	Indicates the bandwidth used while transmitting sound/audio to this user.	Kbps	Comparing these values across users will reveal which user is sending/receiving bandwidth-intensive sound/audio files over the ICA channel.
Audio bandwidth input	Indicates the bandwidth used while receiving sound/audio from this user.	Kbps	
COM bandwidth input	Indicates the bandwidth used when sending data to this user's COM port.	Kbps	Comparing these values across users will reveal which user's COM port is sending/receiving bandwidth-intensive data over the ICA channel.
COM bandwidth ouput	Indicates the bandwidth used when receiving data from this user's COM port.	Kbps	
Drive bandwidth input	Indicates the bandwidth used when this user performs file operations on the mapped drive on the virtual desktop.	Kbps	Comparing the values of these measures across users will reveal which user is performing bandwidth- intensive file operations over the ICA channel.
Drive bandwidth output	Indicates the bandwidth used when the virtual desktop performs file operations on the client's drive.	Kbps	

Measurement	Description	Measurement Unit	Interpretation
Printer bandwidth input	Indicates the bandwidth used when this user prints to a desktop printer over the ICA channel.	Kbps	Comparing the values of these measures across users will reveal which user is issuing bandwidth-intensive print commands over the ICA channel.
Printer bandwidth output	Indicates the bandwidth used when the desktop responds to print jobs issued by this user.	Kbps	If bandwidth consumption is too high, you may want to consider disabling printing. Alternatively, you can avoid printing large documents over the ICA connection.
Session bandwidth input	Indicates the bandwidth used from this user to the virtual desktop for a session	Kbps	Comparing the values of these measures across users will reveal which user and which virtual desktop is performing bandwidth-intensive operatons for a session.
Session bandwidth output	Indicates the bandwidth used from the virtual desktop to this user for a session.	Kbps	
Session compression input	Indicates the compression ratio used from this user to the virtual desktop for a session.	Number	Compression reduces the size of the data that is transacted over the ICA channel. Comparing the values of these measures across users will reveal which client has been configured with a very low and a very high compression ratio.
Session compression output	Indicates the compression ratio used from the virtual desktop to this user for a session.	Number	In the event of high bandwidth usage over an ICA channel, you can set a higher compression ratio for the corresponding client and thus reduce bandwidth consumption.
Speed screen data channel	Indicates the bandwidth used from this user to the virtual desktop for data	Kbps	Comparing the values of these measures across users will reveal

Measurement	Description	Measurement Unit	Interpretation
bandwidth input	channel traffic.		which user has been transmitting/receiving bandwidth-intensive data channel traffic.
Speed screen data channel bandwidth output	Indicates the bandwidth used from virtual desktop to this user for data channel traffic.	Kbps	
Speed screen multimedia acceleration bandwidth input	Indicates the bandwidth used from this user to virtual desktop for multimedia traffic.	Kbps	Comparing the values of these measures across users will reveal which user has been transmitting/receiving bandwidth-intensive multimedia traffic.
Speed screen multimedia acceleration bandwidth output	Indicates the bandwidth used from the virtual desktop to this user for multimedia traffic	Kbps	
HDX media stream for flash data bandwidth input	Indicates the bandwidth used from this user to virtual desktop for flash data traffic.	Kbps	Comparing the values of these measures across users will reveal which user has been transmitting/receiving bandwidth-intensive flash data.
HDX media stream for flash data bandwidth output	Indicates the bandwidth used from the virtual desktop to this user for flash data traffic	Kbps	
USB bandwidth input	Indicates the bandwidth used from this user to the virtual desktop for the USB port-related traffic.	Kbps	Comparing the values of these measures across users will reveal which user has been transmitting/receiving bandwidth-intensive USB traffic.
USB bandwidth	Indicates the bandwidth used from the	Kbps	

Measurement	Description	Measurement Unit	Interpretation
output	virtual desktop to this user for the USB port-related traffic.		
Input line speed	Indicates the average line speed of all the sessions of this user to the desktop.	KB/Sec	
Output line speed	Indicates the average line speed of all sessions from the desktop to this user.	KB/Sec	
Bandwidth usage	Indicates the percentage HDX bandwidth consumption of this user.	Percent	Compare the value of this measure across users to know which user is consuming the maximum HDX bandwidth.

4.2.3 PCoIP Session - VM Test

PCoIP - PC over IP - is a proprietary protocol for remote workstation and desktop resolution. VMware View supports PCoIP to deliver virtual desktops to users connecting to the VDI. Since PCoIP recognizes different types of content and then uses different compression algorithms based on the content type, it is often considered ideal to deliver on the VDI promise of a rich user experience.

The key factors influencing user experience in such cases are the latencies experienced by the user while connecting to the desktop via PCoIP and the bandwidth used when a user interacts with a virtual desktop. High latency and excessive bandwidth consumption can often slowdown access to desktops, thereby significantly delaying subsequent user operations. Hence, monitoring the latency and bandwidth usage of the PCoIP communication channel between the user terminal and the virtual desktops is essential.

The PCoIP Session - VM test auto-discovers the virtual desktops on the ESX host and the users who are currently connected to each desktop. For each such user, the test monitors the communication between a user and the virtual desktop, and reports the following:

- The latency experienced by each user session;
- The bandwidth used by the incoming and outgoing data/audio/multimedia traffic transacted by the PCoIP communication channel between each user and virtual desktop;

Using this test, an administrator can identify user sessions that are being impacted by high latency and abnormal bandwidth usage. In addition, the test also reveals the type of traffic that is causing excessive

bandwidth usage, thereby providing pointers to how the client configuration can be fine-tuned in order to reduce bandwidth consumption and improve performance.

Note:

This test is relevant only where VMware Horizon View is used to broker connections between the user and the desktops. That is why, this test is disabled by default. To enable the test, go to the enable / disable tests page using the menu sequence: Agents -> Tests -> Enable/Disable, pick *Microsoft Hyper-V - VDI* as the **Component type**, set *Performance* as the **Test type**, choose this test from the **DISABLED TESTS** list, and click on the >> button to move the test to the **ENABLED TESTS** list.

Target of the test:

A Hyper-V server hosting virtual desktops

Agent executing the test:

An internal agent

Output of the test:

One set of results will be reported for every user who is connected to a virtual desktop, via PCoIP

Parameters of the test:

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured.
3. **PORT** - The port at which the **HOST** listens. By default, this is NULL.
4. **INSIDE VIEW USING** - By default, this test communicates with every VM remotely and extracts "inside view" metrics. Therefore, by default, the **INSIDE VIEW USING** flag is set to **Remote connection to VM (Windows)**.

   Typically, to establish this remote connection, eG Enterprise requires that the eG agent be configured with domain administrator privileges. In high-security environments, where the IT staff might have reservations about exposing the credentials of their domain administrators, this approach to extracting "inside view" metrics might not be preferred. In such environments therefore, eG Enterprise provides administrators the option to deploy a piece of software called the **eG VM Agent** on every Windows VM; this VM agent allows the eG agent to collect "inside view" metrics from the Windows VMs **without domain administrator rights.** Refer to Section 1.3.2 for more details on the **eG VM Agent**. To ensure that the "inside view" of Windows VMs is obtained using the eG VM Agent, set the **INSIDE VIEW USING** flag to **eG VM Agent (Windows)**. Once this is done, you can set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.

5. **DOMAIN, ADMIN USER, ADMIN PASSWORD,** and **CONFIRM PASSWORD** – By default, this test connects to each virtual guest remotely and attempts to collect "inside view" metrics. Accordingly, the **INSIDE VIEW USING** flag is set to **Remote connection to VM (Windows)** by default. To obtain a remote connection, the test must be configured with the privileges of an administrative user to the domain within which the guests reside. The first step towards this is to specify the **DOMAIN** within which the virtual guests reside. The **ADMIN USER** and **ADMIN PASSWORD** will change according to the **DOMAIN** specification. Discussed below are the different values that the **DOMAIN** parameter can take, and how they impact the **ADMIN USER** and **ADMIN PASSWORD** specifications:

   - **If the VMs belong to a single domain:** If the guests belong to a specific domain, then specify the name of that domain against the domain parameter. In this case, any administrative user in that domain will have remote access to all the virtual guests. Therefore, an administrator account in the given domain can be provided in the **ADMIN USER** field and the corresponding password in the **ADMIN PASSWORD** field. Confirm the password by retyping it in the **CONFIRM PASSWORD** text box.
   - I**f the VMs belong to different domains:**In this case, you might want to provide multiple domain names. If this is done, then, to access the guests in every configured domain, the test should be configured with the required user privileges; this implies that along with multiple **DOMAIN** names, multiple **ADMIN USER** names and **ADMIN PASSWORDS** would also have to be provided. To help administrators provide these user details quickly and easily, the eG administrative interface embeds a special configuration page. To access this page, simply click on the encircled '+' icon that appears alongside the . To know how to use the special page, refer to Section 3.6.1 of this document.
   - **If the INSIDE VIEW USING flag is set to 'eG VM Agent (Windows)'**: On the other hand, if the **INSIDE VIEW USING** flag is set to **eG VM Agent (Windows)**, then it implies that the inside view can be obtained without domain administrator privileges. Therefore, set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.

4. **REPORT BY USER** – For the *Hyper-V* monitoring model, the **REPORT BY USER** flag is set to **NO** by default, indicating that by default, the guest operating systems on the Hyper-V server are identified using the hostname specified in the operating system. On the other hand, for the *Hyper-V VDI* model, this flag is set to **YES** by default; this implies that in case of VDI servers, by default, the guests will be identified using the login of the user who is accessing the guest OS. In other words, in VDI environments, this test will, by default, report measures for every *username_on_virtualmachinename*.

5. **REPORT POWERED OS** - **This flag becomes relevant only if the REPORT BY USER flag is set to 'Yes'.**

   If the **REPORT POWERED OS** flag is set to **Yes** (which is the default setting), then this test will report measures for even those VMs that do not have any users logged in currently. Such guests will be identified by their *virtualmachine name* and not by the *username_on_virtualmachinename*. On the other hand, if the **REPORT POWERED OS** flag is set to **No**, then this test will not report measures for those VMs to which no users are logged in currently.

6. **IGNORE VMS INSIDE VIEW** - Administrators of some high security Hyper-V environments might not have permissions to internally monitor one/more VMs. The eG agent can be configured to **not obtain the 'inside view' of such 'inaccessible' VMs** using the **IGNORE VMS INSIDE VIEW** parameter. Against this parameter, you can provide a comma-separated list of VM names, or VM name patterns,

for which the inside view need not be obtained. For instance, your **IGNORE VMS INSIDE VIEW** specification can be: **xp,*lin*,win*,vista*. Here, the * (asterisk) is used to denote leading and trailing spaces (as the case may be). By default, this parameter is set to *none* indicating that the eG agent obtains the inside view of all VMs on a Hyper-V host by default.

Note:
While performing VM discovery, the eG agent will not discover the operating system of the VMs configured in the **IGNORE VMS INSIDE VIEW** text box.

7. **EXCLUDE VMS** - Administrators of some virtualized environments may not want to monitor some of their less-critical VMs - for instance, VM templates - both from 'outside' and from 'inside'. The eG agent in this case can be configured to completely exclude such VMs from its monitoring purview. To achieve this, provide a comma-separated list of VMs to be excluded from monitoring in the **EXCLUDE VMS** text box. Instead of VMs, VM name patterns can also be provided here in a comma-separated list. For example, your **EXCLUDE VMS** specification can be: **xp,*lin*,win*,vista*. Here, the * (asterisk) is used to denote leading and trailing spaces (as the case may be). By default, this parameter is set to *none* indicating that the eG agent obtains the inside and outside views of all VMs on a virtual host by default. By providing a comma-separated list of VMs/VM name patterns in the **EXCLUDE VMS** text box, you can make sure the eG agent stops collecting 'inside' and 'outside' view metrics for a configured set of VMs.
8. **IGNORE WINNT** – By default, the eG agent does not support the *inside view* for VMs executing on **Windows NT** operating systems. Accordingly, the **IGNORE WINNT** flag is set to **Yes** by default.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
Round trip time	Indicates the round trip latency between the virtual desktop and this user terminal.	Secs	Comparing the value of this measure across users will enable administrators to quickly and accurately identify users who are experiencing higher latency when connecting to a virtual desktop.
Data received rate	Indicates the rate at which data was received by this user from the virtual desktop.	Kbit/Sec	Comparing the value of each of these measures across users will enable administrators to quickly and accurately identify users who are consuming the maximum bandwidth. Once you zero-in on the user, you can compare the *Data*

Measurement	Description	Measurement Unit	Interpretation
			received rate of that user with the *Data sent rate* to know when the user consumed more bandwidth - when receiving data or while sending data?
Data sent rate	Indicates the rate at which data was sent by this user to the virtual desktop.	Kbit/Sec	
Audio data received rate	Indicates the bandwidth used while transmitting sound/audio to this user.	Kbit/Sec	Comparing these values across users will reveal which user is sending/receiving bandwidth-intensive sound/audio files over PCoIP.
Audio data sent rate	Indicates the bandwidth used while receiving sound/audio from this user.	Kbit/Sec	
Imaging data received rate	Indicates the bandwidth used when sending imaging data to this user.	Kbit/Sec	Comparing these values across users will reveal which user is sending/receiving bandwidth-intensive images over PCoIP.
Imaging data sent rate	Indicates the bandwidth used when receiving imaging data from this user.	Kbit/Sec	
Imaging decoder capability rate	Indicates the currrent estimate of the decoder processing capability.	Kbit/Sec	
Incoming bandwidth rate	Indicates the overall bandwidth used by incoming PCoIP packets.	Kbit/Sec	Comparing the values of these measures across users will reveal which user is performing bandwidth-intensive operations over the PCoIP channel.
Outgoing bandwidth rate	Indicates the overall bandwidth used by outgoing PCoIP packets.	Kbit/Sec	
USB data received rate	Indicates the bandwidth used when this user received USB data over the PCoIP channel.	Kbit/Sec	Comparing the values of these measures across users will reveal which user is sending/receiving bandwidth-intensive USB data over the PCoIP channel.
USB data sent rate	Indicates the bandwidth used when this user sent USB data over the PCoIP channel.	Kbit/Sec	

Measurement	Description	Measurement Unit	Interpretation
Received packets lost	Indicates the percentage of packets received by this user that were lost.	Percent	A high value for these measures is indicative of a bad network connection between the user terminal and the virtual desktop.
Transmitted packets lost	Indicates the percentage of packets transmitted by this user that were lost.	Percent	
Imaging encoded frames	Indicates the number of imaging frames that were encoded per second.	Frames/Sec	

4.2.4 User Profile Management – VM Test

User logon is a complex and resource intensive process in a VDI environment, and is a key determinant of the quality of a user's experience with the VDI service. This process is initiated when a desktop broker's load balancing algorithm selects the virtual desktop where a published application or desktop, which a user has selected, will be started and ends when the application or desktop is running and the user is able to interact with it.

Delays in the user logon process can therefore serve as key spoilers of a user's experience with the desktop service, causing significant loss of revenue and reputation in mission-critical VDI environments.

One of the common causes for delays in user logons is a delay in the loading of user profiles. To reduce the time taken to load profiles and thus minimize the user logon time, VDI environments where user connections are brokered through the Citrix XenDesktop Broker, use the Citrix Profile Management solution. Citrix Profile Management is a profile type that supersedes all other profiles for the user.

During logon, the Profile management service manages the user settings in a user profile. This service helps minimize the user logon time by enabling administrators to exclude (and include) certain files and folders in order to prevent extraneous settings from needlessly being copied with the profile. For example, some applications may create folders and files that account for tens or hundreds of megabytes—data that is really not required. By excluding these items, the profile is thus smaller, and smaller profiles load faster. Alternatively, you could elect to only include specific files and folders, thus keeping to a minimum the amount of profile data being managed within the user‘s profile.

Also, upon logoff, the Profile management service merges back only changed user settings to the centrally stored user settings (user‘s store).

In environments where the Citrix Profile Management service is utilized therefore, the user experience with the VDI service greatly depends upon how efficient the service is.

To ascertain the efficiency of the Citrix Profile Management service, VDI administrators may have to periodically track the logon/logoff duration and profile size of each user to the virtual desktops operating on a

target virtual host. Doing so will enable these administrators to determine whether/not the Profile management service has succeeded in minimizing both user logon times and profile sizes. The User Profile Management - Guest test helps administrators perform this check at pre-configured intervals. The 'per-user' performance results reported by this test will not only enable administrators to judge the effectiveness of the Profile management service in its entirety, but will also shed light on those user logons/logoffs that are still experiencing delays; this provides insights into how the service can be fine-tuned to enhance the VDI experience of such users.

Note:

This test is relevant only where the Citrix XenDesktop Broker is used to broker connections between the user and the desktops. This is why, this test is disabled by default. To enable the test, go to the **ENABLE / DISABLE TESTS** page using the menu sequence: Agents -> Tests -> Enable/Disable, pick the *Microsoft Hyper-V - VDI* as the **Component type**, set *Performance* as the **Test type**, choose this test from the **DISABLED TESTS** list, and click on the >> button to move the test to the **ENABLED TESTS** list.

Target of the test:

A Hyper-V server hosting virtual desktops

Agent executing the test:

An internal agent

Output of the test:

One set of results will be reported for every user who is connected to a virtual desktop

Parameters of the test:

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured.
3. **PORT** - The port at which the **HOST** listens. By default, this is NULL.
4. **INSIDE VIEW USING** - By default, this test communicates with every VM remotely and extracts “inside view” metrics. Therefore, by default, the **INSIDE VIEW USING** flag is set to **Remote connection to VM (Windows)**.

   Typically, to establish this remote connection, eG Enterprise requires that the eG agent be configured with domain administrator privileges. In high-security environments, where the IT staff might have reservations about exposing the credentials of their domain administrators, this approach to extracting “inside view” metrics might not be preferred. In such environments therefore, eG Enterprise provides administrators the option to deploy a piece of software called the **eG VM Agent** on every Windows

VM; this VM agent allows the eG agent to collect "inside view" metrics from the Windows VMs **without domain administrator rights.** Refer to Section 1.3.2 for more details on the **eG VM Agent**. To ensure that the "inside view" of Windows VMs is obtained using the eG VM Agent, set the **INSIDE VIEW USING** flag to **eG VM Agent (Windows)**. Once this is done, you can set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.

5. **DOMAIN, ADMIN USER, ADMIN PASSWORD,** and **CONFIRM PASSWORD** – By default, this test connects to each virtual guest remotely and attempts to collect "inside view" metrics. Accordingly, the **INSIDE VIEW USING** flag is set to **Remote connection to VM (Windows)** by default. To obtain a remote connection, the test must be configured with the privileges of an administrative user to the domain within which the guests reside. The first step towards this is to specify the **DOMAIN** within which the virtual guests reside. The **ADMIN USER** and **ADMIN PASSWORD** will change according to the **DOMAIN** specification. Discussed below are the different values that the **DOMAIN** parameter can take, and how they impact the **ADMIN USER** and **ADMIN PASSWORD** specifications:

   - **If the VMs belong to a single domain:** If the guests belong to a specific domain, then specify the name of that domain against the domain parameter. In this case, any administrative user in that domain will have remote access to all the virtual guests. Therefore, an administrator account in the given domain can be provided in the **ADMIN USER** field and the corresponding password in the **ADMIN PASSWORD** field. Confirm the password by retyping it in the **CONFIRM PASSWORD** text box.
   - I**f the VMs belong to different domains:**In this case, you might want to provide multiple domain names. If this is done, then, to access the guests in every configured domain, the test should be configured with the required user privileges; this implies that along with multiple **DOMAIN** names, multiple **ADMIN USER** names and **ADMIN PASSWORDS** would also have to be provided. To help administrators provide these user details quickly and easily, the eG administrative interface embeds a special configuration page. To access this page, simply click on the encircled '+' icon that appears alongside the . To know how to use the special page, refer to Section 3.6.1 of this document.
   - **If the INSIDE VIEW USING flag is set to 'eG VM Agent (Windows)'**: On the other hand, if the **INSIDE VIEW USING** flag is set to **eG VM Agent (Windows)**, then it implies that the inside view can be obtained without domain administrator privileges. Therefore, set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.

4. **REPORT BY USER** – For the *Hyper-V* monitoring model, the **REPORT BY USER** flag is set to **NO** by default, indicating that by default, the guest operating systems on the Hyper-V server are identified using the hostname specified in the operating system. On the other hand, for the *Hyper-V VDI* model, this flag is set to **YES** by default; this implies that in case of VDI servers, by default, the guests will be identified using the login of the user who is accessing the guest OS. In other words, in VDI environments, this test will, by default, report measures for every *username_on_virtualmachinename*.

5. **REPORT POWERED OS** - **This flag becomes relevant only if the REPORT BY USER flag is set to 'Yes'.**

   If the **REPORT POWERED OS** flag is set to **Yes** (which is the default setting), then this test will report measures for even those VMs that do not have any users logged in currently. Such guests will be identified by their *virtualmachine name* and not by the *username_on_virtualmachinename*. On the other hand, if the **REPORT POWERED OS** flag is set to **No**, then this test will not report measures for those

VMs to which no users are logged in currently.

6. **IGNORE VMS INSIDE VIEW** - Administrators of some high security Hyper-V environments might not have permissions to internally monitor one/more VMs. The eG agent can be configured to **not obtain the 'inside view' of such 'inaccessible' VMs** using the **IGNORE VMS INSIDE VIEW** parameter. Against this parameter, you can provide a comma-separated list of VM names, or VM name patterns, for which the inside view need not be obtained. For instance, your **IGNORE VMS INSIDE VIEW** specification can be: **xp,*lin*,win*,vista*. Here, the * (asterisk) is used to denote leading and trailing spaces (as the case may be). By default, this parameter is set to *none* indicating that the eG agent obtains the inside view of all VMs on a Hyper-V host by default.

   Note:
   While performing VM discovery, the eG agent will not discover the operating system of the VMs configured in the **IGNORE VMS INSIDE VIEW** text box.

7. **EXCLUDE VMS** - Administrators of some virtualized environments may not want to monitor some of their less-critical VMs - for instance, VM templates - both from 'outside' and from 'inside'. The eG agent in this case can be configured to completely exclude such VMs from its monitoring purview. To achieve this, provide a comma-separated list of VMs to be excluded from monitoring in the **EXCLUDE VMS** text box. Instead of VMs, VM name patterns can also be provided here in a comma-separated list. For example, your **EXCLUDE VMS** specification can be: **xp,*lin*,win*,vista*. Here, the * (asterisk) is used to denote leading and trailing spaces (as the case may be). By default, this parameter is set to *none* indicating that the eG agent obtains the inside and outside views of all VMs on a virtual host by default. By providing a comma-separated list of VMs/VM name patterns in the **EXCLUDE VMS** text box, you can make sure the eG agent stops collecting 'inside' and 'outside' view metrics for a configured set of VMs.

8. **IGNORE WINNT** – By default, the eG agent does not support the *inside view* for VMs executing on **Windows NT** operating systems. Accordingly, the **IGNORE WINNT** flag is set to **Yes** by default.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
Logon Duration	Indicates the duration of logon processing for this user.	Secs	This value helps to measure the reduction in logon times when the Profile Management service 'streams' the profile. Ideally therefore, this value should be low. A high value or a consistent increase in the value of this measure could indicate that profile loading still takes a lot of time at logon - this could be owing to a

Measurement	Description	Measurement Unit	Interpretation
			large profile size. You can then check the value reported by the *Logon Bytes* measure to know the profile size at logon. If profile sizes continue to grow at logon despite the use of Profile management, it is indicative of the ineffectiveness of profile management. You may then have to fine-tune the feature to further reduce the profile size by excluding more unnecessary files from the profile, or you may have to explore other options such as roaming profiles, mandatory profiles, etc.
Logon Bytes	Indicates the size of this user's profile when it is retrieved from the user's store at logon.	MB	Ideally, the value of this measure should be low. A low profile size could result in faster profile loading at logon, lesser time to login, and consequently, a richer user experience with the VDI service. If profile sizes continue to grow despite the use of Profile management, it is indicative of the ineffectiveness of profile management. You may then have to fine-tune the feature to further reduce the profile size by excluding more unnecessary files from the profile.
Logoff Duration	Indicates the duration of logoff processing for this user.	Secs	A low value is desired for this measure. A high value could indicate that the profile management service takes too long to update the user's store with changes in the user settings. This could be because of a bad network connection between the virtual

Measurement	Description	Measurement Unit	Interpretation
			desktop and the user's store, or because too many changes are waiting to be written to the user store.
Logoff Bytes	Indicates the size of this user's profile when it is copied to the user store at logoff.	MB	This measure provides a fair idea of the volume of changes that were copied to the user's store at logoff.
Local Profile Setup Duration	Indicates the time taken to create or prepare this user's profile on the local computer.	Secs	A low value is desired for these measures. If a user complaints of delays during logon, you can use the value of these measures to determine where the VDI service is spending too much time - is it when setting up the local profile? or is it when deleting the local profile?
Delete Local Profile Duration	Indicates the time spent deleting this user's local profiles during the initial migration.	Secs	
Processed Logon Files Under 1KB	Indicates the number of locally copied files for this user's profile that are synchronized during logon and categorized by the file size of 1KB.	Number	All the **Processed Logon Files** measures help VDI administrators to understand whether/not 'profile streaming' (performed by the Profile Management service) has helped in reducing the number of locally copied files during logon. All the **Processed Logoff Files** measures help VDI administrators to understand how many files changed when the user session was in progress.
Processed Logoff Files Under 1KB	Indicates the number of locally copied file for this user's profile that are synchronized during logoff and categorized by the file size of 1KB.	Number	
Processed Logon Files from 1KB to 10KB	Indicates the number of locally copied files for this user's profile that are synchronized during logon and categorized by the file size ranging from 1KB to 10KB.	Number	
Processed Logoff Files	Indicates the number of locally	Number	

Measurement	Description	Measurement Unit	Interpretation
from 1KB to 10KB	copied files for this user's profile that are synchronized during logoff and categorized by the file size ranging from 1KB to 10KB.		
Processed Logon Files from 10KB to 100KB	Indicates the number of locally copied files for this user's profile that are synchronized during logon and categorized by the file size ranging from 10KB to 100KB.	Number	
Processed Logoff Files from 10KB to 100KB	Indicates the number of locally copied files for this user's profile that are synchronized during logoff and categorized by the file size ranging from 1KB to 10KB.	Number	All the **Processed Logon Files** measures help VDI administrators to understand whether/not 'profile streaming' (performed by the Profile Management service) has helped in reducing the number of locally copied files during logon. All the **Processed Logoff Files** measures help VDI administrators to understand how many files changed when the user session was in progress.
Processed Logon Files from 100KB to 1MB	Indicates the number of locally copied files for this user's profile that are synchronized during logon and categorized by the file size ranging from 100KB to 1MB.	Number	
Processed Logoff Files from 100KB to 1MB	Indicates the number of locally copied files for this user's profile that are synchronized during logoff and categorized by the file size ranging from 100KB to 1MB.	Number	
Processed Logon Files from 1MB to 5MB	Indicates the number of locally copied files for this user's profile that are synchronized during logon and categorized by the file size ranging from 1MB to 5MB.	Number	
Processed Logoff Files from 1MB to 5MB	Indicates the number of locally copied files for this user's profile that are synchronized during	Number	All the **Processed Logon Files** measures help VDI administrators to understand whether/not 'profile

Measurement	Description	Measurement Unit	Interpretation
	logoff and categorized by the file size ranging from 1MB to 5MB.		streaming' (performed by the Profile Management service) has helped in reducing the number of locally copied files during logon. All the **Processed Logoff Files** measures help VDI administrators to understand how many files changed when the user session was in progress.
Processed Logon Files Above 5MB	Indicates the number of locally copied files for this user's profile that are synchronized during logon and categorized by the file size above 5MB.	Number	
Processed Logoff Files Above 5MB	Indicates the number of locally copied files for this user's profile that are synchronized during logoff and categorized by the file size above 5MB.	Number	

4.2.5 Domain Time Sync – VM Test

Time synchronization is one of the most important dependencies of windows. A time protocol is responsible for determining the best available time information and converging the clocks to ensure that a consistent time is maintained across systems. By default, windows support a tolerance of plus or minus five minutes for clocks. If the time variance exceeds this setting, clients will be unable to authenticate and in the case of domain controllers, replication will not occur. It implements a time synchronization system based on Network Time Protocol (NTP).

NTP is a fault-tolerant, highly scalable time protocol and it is used for synchronizing computer clocks by using a designated reference clock. A reference clock is some device or machinery that spits out the current time. The special thing about these things is accuracy. Reference clocks must be accurately following some time standard. NTP will compute some additional statistical values based on the current time reported by the reference clock, which will describe the quality of time it sees. Among these values are: offset (or phase), jitter (or dispersion), frequency error, and stability. Thus each NTP server will maintain an estimate of the quality of its reference clocks and of itself.

This test reports the time difference between the reference clock and that of the target environment, and thus helps assess the quality of time seen by the Windows VM. With the help of this test, you can also easily determine whether the reference time changed recently.

This test is disabled by default.To enable the test, go to the **ENABLE / DISABLE TESTS** page using the menu sequence: Agents -> Tests -> Enable/Disable, pick the *Microsoft Hyper-V - VDI* as the **Component type**, set *Performance* as the **Test type**, choose this test from the **DISABLED TESTS** list, and click on the >> button to move the test to the **ENABLED TESTS** list.

Note:

This test reports metrics for Windows VMs only.

Target of the test:

A Hyper-V server hosting virtual desktops

Agent executing the test:

An internal agent

Output of the test:

One set of results will be reported for every Windows virtual desktop on the monitored Hyper-V server

Parameters of the test:

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured.
3. **PORT** - The port at which the **HOST** listens. By default, this is NULL.
4. **INSIDE VIEW USING** - By default, this test communicates with every VM remotely and extracts "inside view" metrics. Therefore, by default, the **INSIDE VIEW USING** flag is set to **Remote connection to VM (Windows)**.

   Typically, to establish this remote connection, eG Enterprise requires that the eG agent be configured with domain administrator privileges. In high-security environments, where the IT staff might have reservations about exposing the credentials of their domain administrators, this approach to extracting "inside view" metrics might not be preferred. In such environments therefore, eG Enterprise provides administrators the option to deploy a piece of software called the **eG VM Agent** on every Windows VM; this VM agent allows the eG agent to collect "inside view" metrics from the Windows VMs **without domain administrator rights.** Refer to Section 1.3.2 for more details on the **eG VM Agent**. To ensure that the "inside view" of Windows VMs is obtained using the eG VM Agent, set the **INSIDE VIEW USING** flag to **eG VM Agent (Windows)**. Once this is done, you can set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.
5. **DOMAIN, ADMIN USER, ADMIN PASSWORD,** and **CONFIRM PASSWORD** – By default, this test connects to each virtual guest remotely and attempts to collect "inside view" metrics. Accordingly, the **INSIDE VIEW USING** flag is set to **Remote connection to VM (Windows)** by default. To obtain a remote connection, the test must be configured with the privileges of an administrative user to the domain within which the guests reside. The first step towards this is to specify the **DOMAIN** within which the virtual guests reside. The **ADMIN USER** and **ADMIN PASSWORD** will change according to the **DOMAIN** specification. Discussed below are the different values that the **DOMAIN** parameter can take.

Measurement	Description	Measurement Unit	Interpretation
NTP offset	Indicates the time difference between the local clock and the designated reference clock.	Secs	For a tiny offset, NTP will adjust the local clock; for small and larger offsets, NTP will reject the reference time for a while. In the latter case, the operating system's clock will continue with the last corrections effective while the new reference time is being rejected. After some time, small offsets (significantly less than a second) will be slewed (adjusted slowly), while larger offsets will cause the clock to be stepped (set anew). Huge offsets are rejected, and NTP will terminate itself, believing something very strange must have happened.

4.2.6 Browser Activity – VM Test

When a user complains of a virtual desktop slowdown, administrators will have to instantly figure out if that VM is experiencing a resource crunch, and if so, which process/application on the desktop is contributing to it. One of the common reasons for CPU/memory contentions and handle leaks on a virtual desktop is web browsing! If a user to a virtual desktop browses resource-intensive web sites, it is bound to result in over-usage of the resources allocated to that VM, which in turn degrades the performance of not just that VM but even the other VMs on that host. While the **System Details – VM** test can lead administrators to the exact browser application that is consuming the CPU/memory resources of the VM excessively, it does not provide visibility into the precise websites that were been browsed when the resource contention occurred. This is where the **Browser Activity – VM** test helps. For each web browser that is being accessed by a user per virtual desktop, this test reports how every browser uses the allocated CPU, memory, and disk resources and reveals the number and URLs of the web sites that are being accessed using each browser. This way, the test not only points administrators to resource-hungry browsers, but also indicates which web sites were being accessed using that browser.

Note:

- This test will report metrics only if the Windows VM being monitored uses the .Net framework v3.0 (or above).
- This test will not be able to monitor the Microsoft Edge browser on Windows 10 VMs.

Target of the test:

A Hyper-V server hosting virtual desktops

Agent executing the test:

An internal agent

Output of the test

One set of results for each browser used by the currently logged-in user to a Windows virtual desktop

First-level descriptor: VM name

Second-level descriptor: User name

Third-level descriptor: User name

Parameters of the test:

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured.
3. **PORT** - The port at which the **HOST** listens. By default, this is NULL.
4. **INSIDE VIEW USING** - By default, this test communicates with every VM remotely and extracts “inside view” metrics. Therefore, by default, the **INSIDE VIEW USING** flag is set to **Remote connection to VM (Windows)**.

   Typically, to establish this remote connection, eG Enterprise requires that the eG agent be configured with domain administrator privileges. In high-security environments, where the IT staff might have reservations about exposing the credentials of their domain administrators, this approach to extracting “inside view” metrics might not be preferred. In such environments therefore, eG Enterprise provides administrators the option to deploy a piece of software called the **eG VM Agent** on every Windows VM; this VM agent allows the eG agent to collect “inside view” metrics from the Windows VMs **without domain administrator rights.** Refer to Section 1.3.2 for more details on the **eG VM Agent**. To ensure that the “inside view” of Windows VMs is obtained using the eG VM Agent, set the **INSIDE VIEW USING**

flag to **eG VM Agent (Windows)**. Once this is done, you can set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.

5. **DOMAIN, ADMIN USER, ADMIN PASSWORD,** and **CONFIRM PASSWORD** – By default, this test connects to each virtual guest remotely and attempts to collect "inside view" metrics. Accordingly, the **INSIDE VIEW USING** flag is set to **Remote connection to VM (Windows)** by default. To obtain a remote connection, the test must be configured with the privileges of an administrative user to the domain within which the guests reside. The first step towards this is to specify the **DOMAIN** within which the virtual guests reside. The **ADMIN USER** and **ADMIN PASSWORD** will change according to the **DOMAIN** specification. Discussed below are the different values that the **DOMAIN** parameter can take, and how they impact the **ADMIN USER** and **ADMIN PASSWORD** specifications:

   - **If the VMs belong to a single domain:** If the guests belong to a specific domain, then specify the name of that domain against the domain parameter. In this case, any administrative user in that domain will have remote access to all the virtual guests. Therefore, an administrator account in the given domain can be provided in the **ADMIN USER** field and the corresponding password in the **ADMIN PASSWORD** field. Confirm the password by retyping it in the **CONFIRM PASSWORD** text box.
   - **If the VMs belong to different domains:**In this case, you might want to provide multiple domain names. If this is done, then, to access the guests in every configured domain, the test should be configured with the required user privileges; this implies that along with multiple **DOMAIN** names, multiple **ADMIN USER** names and **ADMIN PASSWORDS** would also have to be provided. To help administrators provide these user details quickly and easily, the eG administrative interface embeds a special configuration page. To access this page, simply click on the **Click here** hyperlink that appears just above the parameters of this test in the test configuration page. To know how to use the special page, refer to Section 4.2.6 of this document.
   - **If the INSIDE VIEW USING flag is set to 'eG VM Agent (Windows)':** On the other hand, if the **INSIDE VIEW USING** flag is set to **eG VM Agent (Windows)**, then it implies that the inside view can be obtained without domain administrator privileges. Therefore, set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.

6. **REPORT BY USER** – For the *Microsoft Hyper-V* monitoring model, the **REPORT BY USER** flag is set to **NO** by default, indicating that by default, the guest operating systems on the Hyper-V server are identified using the hostname specified in the operating system. On the other hand, for the *Microsoft Hyper-V VDI* model, this flag is set to **YES** by default; this implies that in case of VDI servers, by default, the guests will be identified using the login of the user who is accessing the guest OS. In other words, in VDI environments, this test will, by default, report measures for every *username_on_virtualmachinename*.
7. **REPORT POWERED OS** - **This flag becomes relevant only if the REPORT BY USER flag is set to 'Yes'.**

   If the **REPORT POWERED OS** flag is set to **Yes** (which is the default setting), then this test will report measures for even those VMs that do not have any users logged in currently. Such guests will be identified by their *virtualmachine name* and not by the *username_on_virtualmachinename*. On the other

hand, if the **REPORT POWERED OS** flag is set to **No**, then this test will not report measures for those VMs to which no users are logged in currently.

8. **REPORT POWERED ON** - You can set the **REPORT POWERED ON** status to **Yes,** so that the test reports an additional measure, *Is VM powered on?*, revealing whether a guest OS is currently running or not. The default status of this flag is set to **Yes** for a *Hyper-V* server. For a *Hyper-V VDI* server on the other hand, the default status of this flag is **No**. This is because, in such environments, the virtual desktops will be in the powered-off state most of the time.
9. **IGNORE VMS INSIDE VIEW** - Administrators of some high security Hyper-V environments might not have permissions to internally monitor one/more VMs. The eG agent can be configured to **not obtain the 'inside view' of such 'inaccessible' VMs** using the **IGNORE VMS INSIDE VIEW** parameter. Against this parameter, you can provide a comma-separated list of VM names, or VM name patterns, for which the inside view need not be obtained. For instance, your **IGNORE VMS INSIDE VIEW** specification can be: **xp,*lin*,win*,vista*. Here, the * (asterisk) is used to denote leading and trailing spaces (as the case may be). By default, this parameter is set to *none* indicating that the eG agent obtains the inside view of all VMs on a Hyper-V host by default.

   Note:

   While performing VM discovery, the eG agent will not discover the operating system of the VMs configured in the **IGNORE VMS INSIDE VIEW** text box.

10. **EXCLUDE VMS** - Administrators of some virtualized environments may not want to monitor some of their less-critical VMs - for instance, VM templates - both from 'outside' and from 'inside'. The eG agent in this case can be configured to completely exclude such VMs from its monitoring purview. To achieve this, provide a comma-separated list of VMs to be excluded from monitoring in the **EXCLUDE VMS** text box. Instead of VMs, VM name patterns can also be provided here in a comma-separated list. For example, your **EXCLUDE VMS** specification can be: **xp,*lin*,win*,vista*. Here, the * (asterisk) is used to denote leading and trailing spaces (as the case may be). By default, this parameter is set to *none* indicating that the eG agent obtains the inside and outside views of all VMs on a virtual host by default. By providing a comma-separated list of VMs/VM name patterns in the **EXCLUDE VMS** text box, you can make sure the eG agent stops collecting 'inside' and 'outside' view metrics for a configured set of VMs.
11. **IGNORE WINNT** – By default, the eG agent does not support the *inside view* for VMs executing on **Windows NT** operating systems. Accordingly, the **IGNORE WINNT** flag is set to **Yes** by default.
12. **DD FREQUENCY** - Refers to the frequency with which detailed diagnosis measures are to be generated for this test. For a *Microsoft Hyper-V* server, this is set to *1:1* by default. This indicates that, by default, detailed measures will be generated every time this test runs, and also every time the test detects a problem. It is recommended that you do not change the default setting of this parameter. This is because, eG Enterprise can discover the IP addresses of the guest operating systems on a Hyper-V host, only while generating the detailed measures for this test. The automatic discovery of the guest IPs, in turn, enables eG Enterprise to perform **AutoVirtualMapping**.

13. **DETAILED DIAGNOSIS** - To make diagnosis more efficient and accurate, the eG suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option.

    The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:

    - The eG manager license should allow the detailed diagnosis capability
    - Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
Running browser instances	Indicates the number of instances of this browser currently running on this virtual desktop.	Number	Use the detailed diagnosis of this measure to know how much resources were utilized by each instance of a browser, so that the resource-hungry instance can be isolated.

Measurement	Description	Measurement Unit	Interpretation
Recent web sites	Indicates the number of websites that were accessed using this browser on this virtual desktop during the last measurement period.	Number	Use the detailed diagnosis of this measure to know which web sites are being accessed using a browser.
CPU utilization	Indicates the percentage CPU usage of this browser on this virtual desktop.	Percent	Compare the value of this measure across browsers to know which browser consumed the maximum CPU on a desktop. If the value of this measure is close to 100% on that desktop, it indicates excessive CPU usage by the browser. You may then want to use the detailed diagnosis of the Recent web sites measure to know which web sites are being accessed using that browser, which caused CPU usage to soar.
Memory used	Indicates the percent usage of memory by this browser on this virtual desktop.	Percent	Compare the value of this measure across browsers to know which browser consumed the maximum memory on a desktop. If the value of this measure is close to 100% on that desktop, it indicates excessive memory usage by the browser. You may then want to use the detailed diagnosis of the Recent web sites measure to know which web sites are being accessed using that browser, which caused CPU usage to soar.
Handles used	Indicates the number of handles opened by this browser on this virtual desktop.	Number	Compare the value of this measure across browsers to

Measurement	Description	Measurement Unit	Interpretation
			know which browser opened the maximum number of handles on a desktop. If the value of this measure consistently increases on that desktop, it indicates that the corresponding browser is leaking memory. You may then want to use the detailed diagnosis of the Recent web sites measure to know which web sites are being accessed using that browser, which caused the memory leak.
Disk reads	Indicates the rate at which this browser read from the disks supported by this virtual desktop.	KB/Sec	A high value for these measures indicates that the browser is generating high disk I/O. You may then want to use the detailed diagnosis of the Recent web sites measure of this browser to know which web sites on the browser are responsible for the high disk I/O.
Disk writes	Indicates the rate at which this browser read from the disks of this virtual desktop.	KB/Sec	
Disk IOPS	Indicates the rate of read and write operations performed by this browser on the disks of this virtual desktop.	Operations/Sec	A high value for this measure indicates that the browser is generating high disk I/O. You may then want to use the detailed diagnosis of the Recent web sites measure of this browser to know which web sites on the browser are responsible for the high disk I/O.
Page faults	Indicates the rate at which page faults by the threads executing in this browser are occurring on this virtual desktop.	Faults/Sec	Ideally, the value of this measure should be low. A high value for a browser is a cause for concern. You may then want to use the detailed diagnosis of the Recent web sites measure of this

Measurement	Description	Measurement Unit	Interpretation
			browser to know which web sites on the browser are responsible for page faults.

The detailed diagnosis of the *Running browser* instances measure reveals the process ID of each browser instance that is currently running on the virtual desktop and the resource usage of each instance. This way, you can easily and accurately identify the instance that is consuming resources excessively.

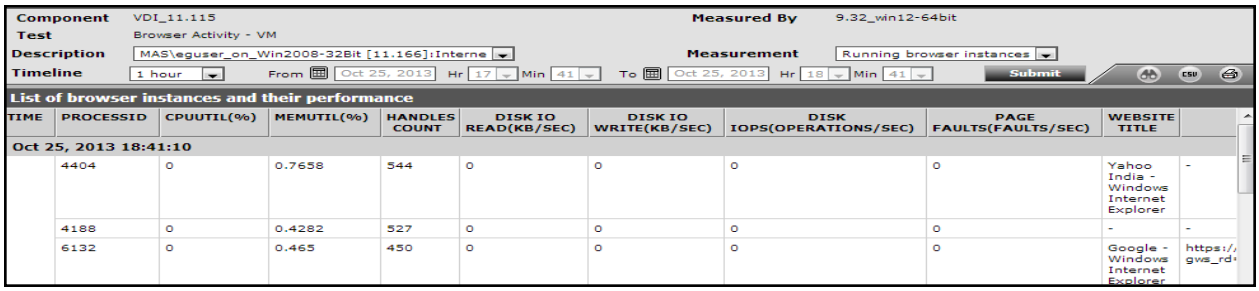


Figure 4.11: The detailed diagnosis of the Running browser instances measure

The detailed diagnosis of the *Recent web sites* measure reveals the names and URLs of the web sites that are being accessed using a browser.

Figure 4.12: The detailed diagnosis of the Recent web sites measure

4.2.7 Personal vDisk – VM Test

The personal vDisk retains the single image management of pooled and streamed desktops while allowing people to install applications and change their desktop settings.

Unlike traditional Virtual Desktop Infrastructure (VDI) deployments involving pooled desktops, where users lose their customizations and personal applications when the administrator alters the base virtual machine (VM), deployments using personal vDisks retain those changes. This means administrators can easily and centrally manage their base VMs while providing users with a customized and personalized desktop experience.

Personal vDisks provide this separation by redirecting all changes made on the user's VM to a separate disk (the personal vDisk) attached to the user's VM. The content of the personal vDisk is blended at runtime with the content from the base VM to provide a unified experience. In this way, users can still access applications provisioned by their administrator in the base VM.

But, what happens if a personal vDisk runs out of space? Simple! Users will no longer be able to hold on to their customizations, allowing them access to only the base VM and the applications installed therein! This outcome beats the entire purpose of having personal vDisks! If this is to be avoided, then administrators should continuously monitor the usage of the personal vDisks, proactively detect a potential space crunch, determine what is causing the rapid erosion of space on the personal vDisk, and fix the root-cause, before desktop users complain. This is where the **Personal vDisk – VM** test helps.

For each VM on a XenServer, this test tracks the status and space usage of its personal vDisk and promptly reports errors / abnormal space usage. This way, administrators can accurately identify personal vDisks with very limited space, which VM such personal vDisks are associated with, and what is consuming too much disk space – user profiles? Or user applications?

Target of the test:

A Hyper-V server hosting virtual desktops

Agent executing the test:

An internal agent

Output of the test

One set of results for the user who is currently connected to each virtual desktop on the monitored server

Parameters of the test:

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured.
3. **PORT** - The port at which the **HOST** listens. By default, this is NULL.
4. **INSIDE VIEW USING** - By default, this test communicates with every VM remotely and extracts “inside view” metrics. Therefore, by default, the **INSIDE VIEW USING** flag is set to **Remote connection to VM (Windows)**.

   Typically, to establish this remote connection, eG Enterprise requires that the eG agent be configured with domain administrator privileges. In high-security environments, where the IT staff might have reservations about exposing the credentials of their domain administrators, this approach to extracting

"inside view" metrics might not be preferred. In such environments therefore, eG Enterprise provides administrators the option to deploy a piece of software called the **eG VM Agent** on every Windows VM; this VM agent allows the eG agent to collect "inside view" metrics from the Windows VMs **without domain administrator rights.** Refer to Section 1.3.2 for more details on the **eG VM Agent**. To ensure that the "inside view" of Windows VMs is obtained using the eG VM Agent, set the **INSIDE VIEW USING** flag to **eG VM Agent (Windows)**. Once this is done, you can set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.

5. **DOMAIN, ADMIN USER, ADMIN PASSWORD,** and **CONFIRM PASSWORD** – By default, this test connects to each virtual guest remotely and attempts to collect "inside view" metrics. Accordingly, the **INSIDE VIEW USING** flag is set to **Remote connection to VM (Windows)** by default. To obtain a remote connection, the test must be configured with the privileges of an administrative user to the domain within which the guests reside. The first step towards this is to specify the **DOMAIN** within which the virtual guests reside. The **ADMIN USER** and **ADMIN PASSWORD** will change according to the **DOMAIN** specification. Discussed below are the different values that the **DOMAIN** parameter can take, and how they impact the **ADMIN USER** and **ADMIN PASSWORD** specifications:

   - **If the VMs belong to a single domain:** If the guests belong to a specific domain, then specify the name of that domain against the domain parameter. In this case, any administrative user in that domain will have remote access to all the virtual guests. Therefore, an administrator account in the given domain can be provided in the **ADMIN USER** field and the corresponding password in the **ADMIN PASSWORD** field. Confirm the password by retyping it in the **CONFIRM PASSWORD** text box.
   - **If the VMs belong to different domains:**In this case, you might want to provide multiple domain names. If this is done, then, to access the guests in every configured domain, the test should be configured with the required user privileges; this implies that along with multiple **DOMAIN** names, multiple **ADMIN USER** names and **ADMIN PASSWORDS** would also have to be provided. To help administrators provide these user details quickly and easily, the eG administrative interface embeds a special configuration page. To access this page, simply click on the **Click here** hyperlink that appears just above the parameters of this test in the test configuration page. To know how to use the special page, refer to Section 4.2.7 of this document.
   - **If the INSIDE VIEW USING flag is set to 'eG VM Agent (Windows)':** On the other hand, if the **INSIDE VIEW USING** flag is set to **eG VM Agent (Windows)**, then it implies that the inside view can be obtained without domain administrator privileges. Therefore, set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.

6. **REPORT BY USER** – For the *Microsoft Hyper-V* monitoring model, the **REPORT BY USER** flag is set to **NO** by default, indicating that by default, the guest operating systems on the Hyper-V server are identified using the hostname specified in the operating system. On the other hand, for the *Microsoft Hyper-V VDI* model, this flag is set to **YES** by default; this implies that in case of VDI servers, by default, the guests will be identified using the login of the user who is accessing the guest OS. In other words, in VDI environments, this test will, by default, report measures for every *username_on_virtualmachinename*.
7. **REPORT POWERED OS** - **This flag becomes relevant only if the REPORT BY USER flag is set to**

'Yes'.

If the **REPORT POWERED OS** flag is set to **Yes** (which is the default setting), then this test will report measures for even those VMs that do not have any users logged in currently. Such guests will be identified by their *virtualmachine name* and not by the *username_on_virtualmachinename*. On the other hand, if the **REPORT POWERED OS** flag is set to **No**, then this test will not report measures for those VMs to which no users are logged in currently.

8. **REPORT POWERED ON** - You can set the **REPORT POWERED ON** status to **Yes,** so that the test reports an additional measure, *Is VM powered on?*, revealing whether a guest OS is currently running or not. The default status of this flag is set to **Yes** for a *Hyper-V* server. For a *Hyper-V VDI* server on the other hand, the default status of this flag is **No**. This is because, in such environments, the virtual desktops will be in the powered-off state most of the time.
9. **IGNORE VMS INSIDE VIEW** - Administrators of some high security Hyper-V environments might not have permissions to internally monitor one/more VMs. The eG agent can be configured to **not obtain the 'inside view' of such 'inaccessible' VMs** using the **IGNORE VMS INSIDE VIEW** parameter. Against this parameter, you can provide a comma-separated list of VM names, or VM name patterns, for which the inside view need not be obtained. For instance, your **IGNORE VMS INSIDE VIEW** specification can be: **xp,*lin*,win*,vista*. Here, the * (asterisk) is used to denote leading and trailing spaces (as the case may be). By default, this parameter is set to *none* indicating that the eG agent obtains the inside view of all VMs on a Hyper-V host by default.

   Note:

   While performing VM discovery, the eG agent will not discover the operating system of the VMs configured in the **IGNORE VMS INSIDE VIEW** text box.

10. **EXCLUDE VMS** - Administrators of some virtualized environments may not want to monitor some of their less-critical VMs - for instance, VM templates - both from 'outside' and from 'inside'. The eG agent in this case can be configured to completely exclude such VMs from its monitoring purview. To achieve this, provide a comma-separated list of VMs to be excluded from monitoring in the **EXCLUDE VMS** text box. Instead of VMs, VM name patterns can also be provided here in a comma-separated list. For example, your **EXCLUDE VMS** specification can be: **xp,*lin*,win*,vista*. Here, the * (asterisk) is used to denote leading and trailing spaces (as the case may be). By default, this parameter is set to *none* indicating that the eG agent obtains the inside and outside views of all VMs on a virtual host by default. By providing a comma-separated list of VMs/VM name patterns in the **EXCLUDE VMS** text box, you can make sure the eG agent stops collecting 'inside' and 'outside' view metrics for a configured set of VMs.
11. **IGNORE WINNT** – By default, the eG agent does not support the *inside view* for VMs executing on **Windows NT** operating systems. Accordingly, the **IGNORE WINNT** flag is set to **Yes** by default.
12. **DD FREQUENCY** - Refers to the frequency with which detailed diagnosis measures are to be generated for this test. For a *Microsoft Hyper-V* server, this is set to *1:1* by default. This indicates that, by default, detailed measures will be generated every time this test runs, and also every time the test detects a

problem. It is recommended that you do not change the default setting of this parameter. This is because, eG Enterprise can discover the IP addresses of the guest operating systems on a Hyper-V host, only while generating the detailed measures for this test. The automatic discovery of the guest IPs, in turn, enables eG Enterprise to perform **AutoVirtualMapping**.

13. **DETAILED DIAGNOSIS** - To make diagnosis more efficient and accurate, the eG suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option.

    The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:

    - The eG manager license should allow the detailed diagnosis capability
    - Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
Personal vDisk service status	Indicates whether Citrix Personal vDisk *service is running or not on this VM.*		The values that this measure can report and their corresponding numeric values have been discussed in the table below: Measure Value / Numeric Value Stopped / 0 Running / 1 Not installed / 2

Measurement	Description	Measurement Unit	Interpretation
			Note: By default, this test reports the **Measures Value**s listed in the table above to indicate the status of the Personal vDisk service. In the graph of this measure however, the same will be represented using the numeric equivalents.
Recompose status	Indicates the status of the initially provisioned disk or the updated image.	Number	Use the detailed diagnosis of this measure to know for which VM the initial personal vDisk provisioning or image update were unsuccessful and why. The VM can be in one of the following states: • **OK** – The initial provisioning or last image update was successful. • **Disk Init** – This is the first time that the personal vDisk has started or been resized. It is being initialized and partitioned by the service. • **Disk Format** – The personal vDisk is being formatted. • **Updating** – The initial provisioning or an image update is in progress. • **Error (Disk Discovery)** – An error state. An error occurred while discovering the personal vDisk. • **Error (Disk Init)** – An error state. An error occurred while partitioning or formatting the personal vDisk. • **Error (Sys Init)** – An error state. An error occurred while starting the Personal vDisk Service or configuring the personal vDisk. • **Error (Update)** – An error state. An error occurred during the initial provisioning or

Measurement	Description	Measurement Unit	Interpretation
			the last image update. • **Unknown** – An error state. An error occurred but the cause is unknown.
Space used by user applications	Indicates the amount of space used by applications installed on the personal vDisk attached to this VM.	MB	Personal vDisks have two parts, which use different drive letters and are by default equally sized. One part comprises a Virtual Hard Disk file (a .vhd file). This contains items such as applications installed in C:\Program Files. By default, this part uses drive V: but is

Measurement	Description	Measurement Unit	Interpretation
			hidden from users. These measures indicate how much space has been allocated to this .vhd file and how much of the allocated space has been utilized by user applications contained in this file. A high value for the *Space used by user applications* and *Space utilized by user applications* measures is indicative of excessive space used by user applications. You can compare the value of these measures across VMs to know which user to which VM has utilized too much space reserved for user applications on the personal vDisk. If the value of the *Space utilized by user applications* measure grows close to 100% for any VM, it implies that potentially, the user to that VM will not be able to install any applications on the personal vDisk; nor access any applications.
Space allocated for user applications:	Indicates the amount of space allocation for storing user applications on the personal vDisk attached to this VM.	MB	
Space utilized by user applications	Indicates the percentage of allocated space used by applications installed on the personal vDisk attached to this VM.	Percent	
Space used by user profiles	Indicates the amount of space used for storing user profiles on the personal vDisk attached to this VM.	MB	Personal vDisks have two parts, which use different drive letters and are by default equally sized. One part comprises C:\Users (in Windows 7) or C:\Documents and Settings (in Windows XP). This contains user data, documents, and the user profile. By default this uses drive P:. These measures indicate how much space has been allocated to user profiles and how much of the allocated space has been utilized by user profiles. A high value for the *Space used by user profiles* and *Space utilized by user profiles* measures is indicative of excessive space

Measurement	Description	Measurement Unit	Interpretation
			used by user profiles. You can compare the value of these measures across VMs to know which VM's user profiles are consuming the maximum space on the personal vDisk. If the value of the *Space utilized by user profiles* measure grows close to 100% for any VM, it implies that potentially, the user to that VM will not be able to store/access any more documents or user data on the personal vDisk .
Space allocated for user profiles	Indicates the amount of space allocated for storing user profiles on the personal vDisk attached to this VM.	MB	
Space utilized by user profiles	ndicates the percentage of allocated space that has been used up by user profiles on the personal vDisk attached to this VM.	Percent	
Free space	Indicates the amount of unused space on the personal vDisk attached to this VM.	MB	Ideally, the value of this measure should be high. You can compare the value of this measure across VMs to know which VM's personal vDisk has the least free space. You may then want to resize that personal vDisk to accommodate more data.
Total size	Indicates the total size of the personal vDisk attached to this VM.	MB	The minimum size of a Personal vDisk is 3 GB, however a size of 10 GB is recommended.
Space utilized	Indicates the percentage of space in the personal vDisk attached to this VM that is currently used.	Percent	A consistent increase in the value of this measure is a cause for concern, as it indicates a gradual erosion of free space in the personal vDisk of a VM. By comparing the value of this measure across VMs, you can identify which VM's personal vDisk is running out of space! Once the VM with the space-hungry vDisk is isolated, you may want to compare the value of the Space utilized by user applications and Space utilized by user profiles measures of that VM, to clearly

Measurement	Description	Measurement Unit	Interpretation
			understand what is occupying too much space in the personal vDisk – is it the user profiles? Or is it the user applications? Based on this inference, you can figure out which drive partition of the personal vDisk has limited free space, and can decide between freeing up space in that partition or allocating more space to the personal vDisk itself.

4.2.8 Virtual Desktop Session Start-up Details Test

Figure 4.13 depicts a typical user logon process to a virtual desktop via XenDesktop broker.

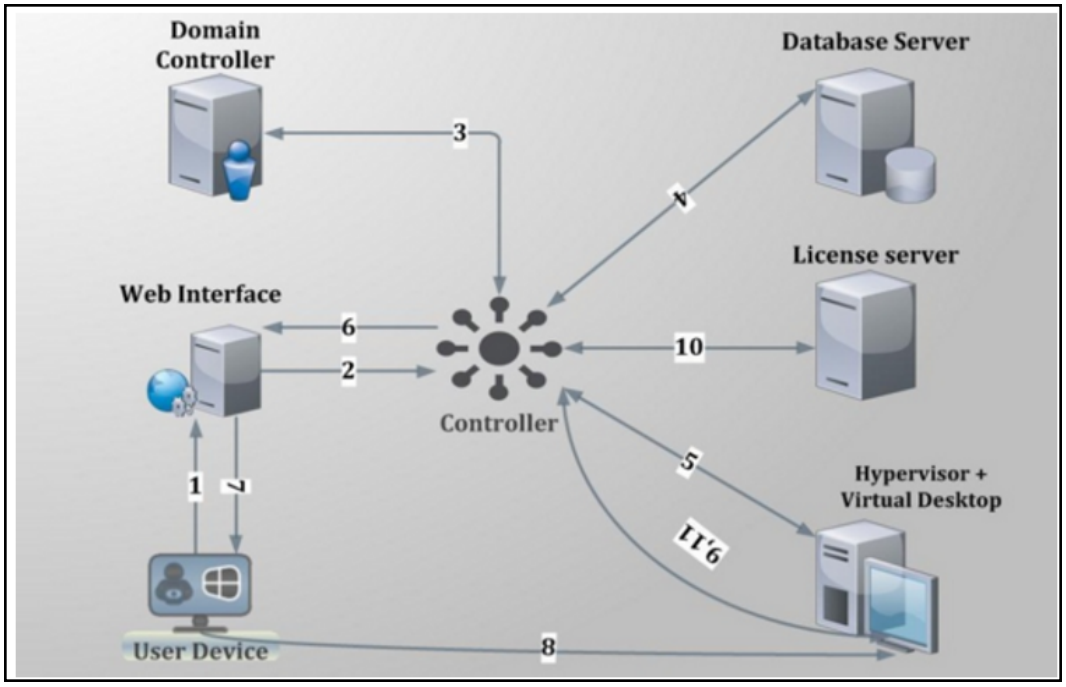


Figure 4.13: Citrix user logon process

The process depicted by Figure 4.13 above has been described below:

1. User provides his/her credentials to the web interface.
2. Web interface forwards the credentials to controller for verification process.

3. Delivery controller transfers these credentials to the domain controller to check if the user is present in the active directory.
4. Once it gets the successful confirmation from AD then controller communicates with site database to check what type of virtual desktop is available for current user.
5. Controller then interacts with the hypervisor layer to gather information about the availability of virtual desktop.
6. Controller then passes the ICA file for user and all the connection information is present inside ICA file so that client can establish the connection.
7. After all the process is complete, the user is assigned the virtual desktop.
8. The user then establishes a connection with the assigned virtual desktop.
9. The virtual desktop again communicates with controller for verification of licensing.
10. Controller checks for license from license server about what type of license is available for user in this current session. License server then communicates back with controller providing the licensing information.
11. Information obtained from license server is then passed to the virtual desktop.

From the discussion above, it can be inferred that login processing happens at two different places – at the delivery controller, and inside the virtual desktop. While login, authentication, and application brokering happen on the delivery controller, session creation and setup happens inside the virtual desktop. A problem in any of these places can result in a poor user experience. Inevitably, these issues result in service desk calls and complaints that "Citrix is slow." Diagnosing login problems has traditionally been a difficult, time-consuming, manual process due to the large number of steps involved. The key to resolving user experience issues therefore, lies in tracking each user's sessions end-to-end, ascertaining the time spent by the session at each step of the logon process – be it on the delivery controller or on the virtual desktop– and accurately identifying where and at what step of the logon process, the slowdown occurred.

To determine the time taken by the entire logon process of a user, isolate logon slowness, and understand where the process was bottlenecked – whether on the delivery controller or on the XenApp server – use the **User Logon Performance** test mapped to the Citrix XA/XD Site component. If the **User Logon Performance** test reveals a problem in session start-up on the virtual desktop, then use the **Virtual Desktop Session Start-up Details** test.

With the **Virtual Desktop Session Start-up Details** test, administrators can receive deep visibility into the virtual desktop end of the Citrix logon process. This test takes an administrator into the virtual desktop, reveals the users who are currently logged on to the virtual desktop, and accurately reports the average time it took for the sessions of each user to start inside the virtual desktop. This way, administrators can rapidly identify which user's sessions are experiencing undue start-up delays.

In addition, the test also provides a break-up of the session start-up duration. This way, the test precisely pinpoints where the delay occurred - – when user credentials were obtained? when credentials were validated? during profile loading? during login script execution? when mapping drives or creating printers?

For this purpose, the test categorizes its metrics into *client start-up metrics and server start-up metrics*.

The *client start-up metrics* are concerned with timing the operations that occur from the point when the user requests for access to a virtual desktop to the point at which a connection to the virtual desktop is established. While onnection-brokering mechanisms involve components that are not on the physical client device, the tasks these systems perform have a direct impact on the performance of the connection start-up and are recorded as part of the client-side process.

The *server start-up metrics* are concerned with timing the operations that occur when creating a new session on the virtual desktop. This includes user authentication, client device mapping, profile loading, login scripts execution, and finally, starting the user's desktop.

Note:

This test will report metrics for only those users who are accessing virtual desktops via a XenDesktop broker.

Target of the test:

A Hyper-V server hosting virtual desktops

Agent executing the test:

An internal agent

Output of the test

One set of results for the user who is currently connected to each virtual desktop on the monitored server

Parameters of the test:

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured.
3. **PORT** - The port at which the **HOST** listens. By default, this is NULL.
4. **INSIDE VIEW USING** - By default, this test communicates with every VM remotely and extracts “inside view” metrics. Therefore, by default, the **INSIDE VIEW USING** flag is set to **Remote connection to VM (Windows)**.

   Typically, to establish this remote connection, eG Enterprise requires that the eG agent be configured with domain administrator privileges. In high-security environments, where the IT staff might have reservations about exposing the credentials of their domain administrators, this approach to extracting “inside view” metrics might not be preferred. In such environments therefore, eG Enterprise provides administrators the option to deploy a piece of software called the **eG VM Agent** on every Windows VM; this VM agent allows the eG agent to collect “inside view” metrics from the Windows VMs **without**

domain administrator rights. Refer to Section 1.3.2 for more details on the **eG VM Agent**. To ensure that the "inside view" of Windows VMs is obtained using the eG VM Agent, set the **INSIDE VIEW USING** flag to **eG VM Agent (Windows)**. Once this is done, you can set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.

5. **DOMAIN, ADMIN USER, ADMIN PASSWORD,** and **CONFIRM PASSWORD** – By default, this test connects to each virtual guest remotely and attempts to collect "inside view" metrics. Accordingly, the **INSIDE VIEW USING** flag is set to **Remote connection to VM (Windows)** by default. To obtain a remote connection, the test must be configured with the privileges of an administrative user to the domain within which the guests reside. The first step towards this is to specify the **DOMAIN** within which the virtual guests reside. The **ADMIN USER** and **ADMIN PASSWORD** will change according to the **DOMAIN** specification. Discussed below are the different values that the **DOMAIN** parameter can take, and how they impact the **ADMIN USER** and **ADMIN PASSWORD** specifications:

   - **If the VMs belong to a single domain:** If the guests belong to a specific domain, then specify the name of that domain against the domain parameter. In this case, any administrative user in that domain will have remote access to all the virtual guests. Therefore, an administrator account in the given domain can be provided in the **ADMIN USER** field and the corresponding password in the **ADMIN PASSWORD** field. Confirm the password by retyping it in the **CONFIRM PASSWORD** text box.
   - **If the VMs belong to different domains:**In this case, you might want to provide multiple domain names. If this is done, then, to access the guests in every configured domain, the test should be configured with the required user privileges; this implies that along with multiple **DOMAIN** names, multiple **ADMIN USER** names and **ADMIN PASSWORDS** would also have to be provided. To help administrators provide these user details quickly and easily, the eG administrative interface embeds a special configuration page. To access this page, simply click on the **Click here** hyperlink that appears just above the parameters of this test in the test configuration page. To know how to use the special page, refer to Section 4.2.8 of this document.
   - **If the INSIDE VIEW USING flag is set to 'eG VM Agent (Windows)':** On the other hand, if the **INSIDE VIEW USING** flag is set to **eG VM Agent (Windows)**, then it implies that the inside view can be obtained without domain administrator privileges. Therefore, set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.

6. **REPORT BY USER** – For the *Microsoft Hyper-V* monitoring model, the **REPORT BY USER** flag is set to **NO** by default, indicating that by default, the guest operating systems on the Hyper-V server are identified using the hostname specified in the operating system. On the other hand, for the *Microsoft Hyper-V VDI* model, this flag is set to **YES** by default; this implies that in case of VDI servers, by default, the guests will be identified using the login of the user who is accessing the guest OS. In other words, in VDI environments, this test will, by default, report measures for every *username_on_virtualmachinename*.
7. **REPORT POWERED OS** - **This flag becomes relevant only if the REPORT BY USER flag is set to 'Yes'.**

   If the **REPORT POWERED OS** flag is set to **Yes** (which is the default setting), then this test will report

measures for even those VMs that do not have any users logged in currently. Such guests will be identified by their *virtualmachine name* and not by the *username_on_virtualmachinename*. On the other hand, if the **REPORT POWERED OS** flag is set to **No**, then this test will not report measures for those VMs to which no users are logged in currently.

8. **REPORT POWERED ON** - You can set the **REPORT POWERED ON** status to **Yes,** so that the test reports an additional measure, *Is VM powered on?*, revealing whether a guest OS is currently running or not. The default status of this flag is set to **Yes** for a *Hyper-V* server. For a *Hyper-V VDI* server on the other hand, the default status of this flag is **No**. This is because, in such environments, the virtual desktops will be in the powered-off state most of the time.
9. **IGNORE VMS INSIDE VIEW** - Administrators of some high security Hyper-V environments might not have permissions to internally monitor one/more VMs. The eG agent can be configured to **not obtain the 'inside view' of such 'inaccessible' VMs** using the **IGNORE VMS INSIDE VIEW** parameter. Against this parameter, you can provide a comma-separated list of VM names, or VM name patterns, for which the inside view need not be obtained. For instance, your **IGNORE VMS INSIDE VIEW** specification can be: **xp,*lin*,win*,vista*. Here, the * (asterisk) is used to denote leading and trailing spaces (as the case may be). By default, this parameter is set to *none* indicating that the eG agent obtains the inside view of all VMs on a Hyper-V host by default.

   Note:

   While performing VM discovery, the eG agent will not discover the operating system of the VMs configured in the **IGNORE VMS INSIDE VIEW** text box.

10. **EXCLUDE VMS** - Administrators of some virtualized environments may not want to monitor some of their less-critical VMs - for instance, VM templates - both from 'outside' and from 'inside'. The eG agent in this case can be configured to completely exclude such VMs from its monitoring purview. To achieve this, provide a comma-separated list of VMs to be excluded from monitoring in the **EXCLUDE VMS** text box. Instead of VMs, VM name patterns can also be provided here in a comma-separated list. For example, your **EXCLUDE VMS** specification can be: **xp,*lin*,win*,vista*. Here, the * (asterisk) is used to denote leading and trailing spaces (as the case may be). By default, this parameter is set to *none* indicating that the eG agent obtains the inside and outside views of all VMs on a virtual host by default. By providing a comma-separated list of VMs/VM name patterns in the **EXCLUDE VMS** text box, you can make sure the eG agent stops collecting 'inside' and 'outside' view metrics for a configured set of VMs.
11. **IGNORE WINNT** – By default, the eG agent does not support the *inside view* for VMs executing on **Windows NT** operating systems. Accordingly, the **IGNORE WINNT** flag is set to **Yes** by default.
12. **DD FREQUENCY** - Refers to the frequency with which detailed diagnosis measures are to be generated for this test. For a *Microsoft Hyper-V* server, this is set to *1:1* by default. This indicates that, by default, detailed measures will be generated every time this test runs, and also every time the test detects a problem. It is recommended that you do not change the default setting of this parameter. This is because, eG Enterprise can discover the IP addresses of the guest operating systems on a Hyper-V host, only while generating the detailed measures for this test. The automatic discovery of the guest

IPs, in turn, enables eG Enterprise to perform **AutoVirtualMapping**.

13. **DETAILED DIAGNOSIS** - To make diagnosis more efficient and accurate, the eG suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option.

    The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:

    - The eG manager license should allow the detailed diagnosis capability
    - Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
User sessions	Indicates the number of sessions currently open for this user on this virtual desktop.	Number	Use the detailed diagnosis of this measure to view the complete details of this user's session. Such details includes the name and IP address of the client from which the session was launched, when session creation started, and when it ended. With the help of this information, administrators

Measurement	Description	Measurement Unit	Interpretation
			can quickly understand if the session took too long to get created.
Session start- up duration:	Indicates the time taken by this user to complete session start-up inside this virtual desktop.	Secs	Compare the value of this measure across users to know which user's sessions took the longest to start on the virtual desktop. To know what is causing this 'slowness', compare the values reported by all the other 'duration' measures of this test for that user on that virtual desktop. This will quickly lead you to where that user's session start- up is spending the maximum time.
Profile load duration	Indicates the time taken to load this user's profile.	Secs	If the user's *Session start-up duration* is high, you may want to compare the value of this measure with that of the other 'duration' measures reported for this user to figure out if a delay in profile loading is what is really ailing that user's logon experience with this virtual desktop. One of the common reasons for high profile load time is the large size of the user profile.
Group policy processing duration	Indicates the time taken by this user's session to process group policies.	Secs	If a user's *Session start-up duration* is high, you may want to compare the value of this measure with that of the other 'duration' measures reported for this user to figure out if a delay in group policy processing is what is really ailing that user's logon experience with this virtual desktop. In such a case, you can also use the detailed diagnosis of this measure to figure out t he names of the group policy client-side extensions (CSE),

Measurement	Description	Measurement Unit	Interpretation
			the time each CSE took to run, the status of every CSE, and errors (if any) encountered by each CSE. Using these in- depth metrics, Citrix administrators can accurately pinpoint which CSE is impeding speedy group policy processing. Note: Detailed diagnostics will be available for this measure only if the **eG VM Agent** is deployed on the virtual desktops and the **INSIDE VIEW USING** parameter of this test is set to eG VM Agent. Logon performance improves when fewer Group Policies are applied. Merge GPOs when possible instead of having multiple GPOs.
Login script execution duration	Indicates the time taken for the login script to execute for this user.	Secs	If a user complains of slowness, then, you can compare the value of this measure with that of the other 'duration' measures of that user to figure out what could have really caused the slowness.
Start- up client duration	This is the high-level client-side connection start- up metric. It starts at the time of the request (mouse click) and ends when the connection between this user's client device and the virtual desktop has been established.	Secs	When any user complains of slowness when trying to logon to a virtual desktop, you may want to compare the value of this measure with that of the *Session start-up server duration* measure of that user to know whether a client-side issue or a server-side issue is responsible for the slowness he/she is experiencing with that virtual desktop. If this comparison reveals that the

Measurement	Description	Measurement Unit	Interpretation
			Start-up client duration of the user is high, it indicates a client-side issue that is causing long start times. In this case therefore, compare the value of the *client start-up* metrics such as the *Application enumeration client duration*, *Configuration file download client duration*, *Credentials obtention client duration*, *ICA file download client duration*, *Launch page web server duration*, *Name resolution client duration*, *Name resolution web server duration*, *Session look-up client duration*, *Session creation client duration*, and *Ticket response web server duration* to know what client-side issue is causing the *Start-up client duration* to be high.
Back-up URL client count	**This measure is relevant when the Citrix Receiver is the session launch mechanism.** It records the number of back-up URL retries before a successful launch. Note that this is the only start-up metric that is a measure of attempts, rather than time duration.	Number	If this metric has a value higher than 1, it indicates that the Web Interface server is unavailable and the Citrix Receiver is attempting to connect to back-up Web Interface servers to launch the virtual desktop. **A value of 2** means that the main Web Interface server was unavailable, but the Citrix Receiver managed to launch the virtual desktop successfully using the first back-up server that it tried. **A value higher than 2** means that multiple Web Interface servers are unavailable. Probable reasons for the non-availability of the Web Interface servers include (in order of likelihood): • Network issues between the client and the server. So the administrator should make sure

Measurement	Description	Measurement Unit	Interpretation
			that the Web Interface server is on the network and accessible to the clients. • An overloaded Web Interface server that is not responding (or has crashed for another reason). Try to log on to the server and check the Windows Performance Monitor/Task Manager to see how much memory is in use and so on. Also, review the Event Logs to see if Windows logged any serious errors.
Application enumeration client duration	**This measure is relevant when the Citrix Receiver is the session launch mechanism.** It measures the time needed by this user's session to retrieve the list of applications from the Web Interface service.	Secs	If the *Start-up client duration* measure reports a high value for a user, then compare the value of this measure with that of the other *client- side* metrics such as *Configuration file download client duration*, *Credentials obtention client duration* , *ICA file download client duration*, *Launch page web server duration*, *Name resolution client duration*, *Name resolution web server duration*, *Session look-up client duration* , *Session creation client duration*, and *Ticket response web server duration* to know whether/not slowness in application enumeration is the precise reason why it took the user a long time to establish a session with the virtual desktop.
Configuration file download client duration	**This measure is relevant when the Citrix Receiver is the session launch mechanism.** It measures the time this user's session took to retrieve the configuration file from the XML broker.	Secs	If the *Start-up client duration* measure reports a high value for a user, then compare the value of this measure with that of the other *client- side* metrics such as *Application enumeration client duration* ,

Measurement	Description	Measurement Unit	Interpretation
			Credentials obtention client duration, *ICA file download client duration*, *Launch page web server duration*, *Name resolution client duration*, *Name resolution web server duration*, *Session look- up client duration*, *Session creation client duration*, and *Ticket response web server duration* to know whether/not slowness in retrieving the configuration file from the XML server is the precise reason why it took the user a long time an ICA session with the XenApp server.
Credentials obtention client duration	**This measure is relevant when the Citrix Receiver is the session launch mechanism.** It measures the time required by this user's session to obtain the user credentials.		Note that COCD is only measured when the credentials are entered manually by the user. Because this metric may be artificially inflated if a user fails to provide credentials in a timely manner, it is subtracted from the *Start-up client duration*. However, in the event that the user manually inputs the credentials, and the value of this measure is higher than that of all the other *client start-up* metrics that this test reports, it is a clear indicator that any connection delay that the user may have experienced is owing to slowness in obtaining user credentials.
ICA file download duration	**This measure is relevant when the Citrix Receiver is the session launch mechanism.** This is the time it takes for this user's client to download the ICA file from the web server.	Secs	The overall process here is: 1. The user clicks on application icon. 2. The user's browser requests the Web Interface launch page. 3. The Web Interface launch page receives the request and starts

Measurement	Description	Measurement Unit	Interpretation
			to process the launch, communicating with the virtual desktop and potentially other components such as Secure Ticket Authority (STA). 4. The Web Interface generates ICA file data. 5. The Web Interface sends the ICA file data back to the user's browser. 6. The browser passes ICA file data to the client. This measure represents the time it takes for the complete process (step 1 to 6). The measure stops counting time when the client receives the ICA file data. The *Launch page web server duration* measure on the other hand, covers the Web server portion of the process (that is, steps 3 and 4). If the *ICA file download duration* is high, but the *Launch page web server duration* is normal, it implies that the server-side processing of the launch was successful, but there were communication issues between the client device and the Web server. Often, this results from network trouble between the two machines, so investigate potential network issues first.
Launch page web server duration	**This measure is relevant when the Web Interface is the session launch mechanism.** It measures the time needed by this user's session to process	Secs	If the value of this measure is high, it indicates at a bottleneck on the Web Interface server. Possible causes include:

Measurement	Description	Measurement Unit	Interpretation
	the launch page (launch. aspx) on the Web Interface server.		• High load on the Web Interface server. Try to identify the cause of the slow down by checking the Internet Information Services (IIS) logs and monitoring tools, Task Manager, Performance Monitor and so on. • Web Interface is having issues communicating with the other components. Check to see if the network connection between Web Interface and virtual desktop is slow. If the Web server seems okay, consider reviewing the virtual desktop for problems.
Name resolution client duration	This is the time it takes the XML service to resolve the name of a published application to an IP address.	Secs	This metric is collected when a client device directly queries the XML Broker to retrieve published application information stored in IMA. This measure is only gathered for new sessions since session sharing occurs during startup if a session already exists. When this metric is high, it indicates the XML Broker is taking a lot of time to resolve the name of a published application to an IP address. Possible causes include a problem on the client, issues with the XML Broker, such as the XML Broker being overloaded, a problem with the network link between the two, or a problem in IMA. Begin by evaluating traffic on the network and the XML Broker.
Name resolution web server	**This measure is relevant when the Citrix Receiver is the**	Secs	When this metric is high, there could be an issue with the Web Interface

Measurement	Description	Measurement Unit	Interpretation
duration	**session launch mechanism.** It is the time it takes the XML service to resolve the name of this virtual desktop to its IP address.		server or the Citrix Receiver, the XML Service, the network link between the two, or a problem in IMA. Like the *Name resolution client duration* measure, this metric indicates how long it takes the XML service to resolve the name of a virtual desktop to its IP address. However, this metric is collected when a Web Interface site is performing this process on behalf of a launch request it has received from either the Citrix Receiver or from a user clicking a Web Interface page icon.
Session look- up client duration	Indicates the time this user's session takes to query every ICA session to host the requested published application.	MSecs	The check is performed on the client to determine whether the application launch request can be handled by an existing session. A different method is used depending on whether the session is new or shared.
Session creation client duration	Indicates the new session creation time.	Secs	In the event of slowness, if the *Start-up client duration* of a user session is found to be higher than the *Session start-up server duration*, you may want to compare the value of this measure with all other *client start-up* measures to determine whether/not session creation is the process that is slowing down the application launch.
Ticket response web server duration	**This measure is relevant when the Citrix Receiver or Web Interface is the desktop launch mechanism.** This is the time this user's sessions take to get a ticket (if required) from the STA server or XML service.	Secs	When this metric is high, it can indicate that the Secure Ticket Authority (STA) server or the XML Broker are overloaded.

Measurement	Description	Measurement Unit	Interpretation
Reconnect enumeration client duration	**This measure is relevant when the Citrix Receiver or Web Interface is the desktop launch mechanism.** This is the time it takes this user's client to get a list of reconnections.	Secs	Compare the value of this measure with that of other *client start-up* metrics for a user to know what is the actual cause for the client start-up delay.
Reconnect enumeration web server duration	**This measure is relevant when the Citrix Receiver or Web Interface is the desktop launch mechanism.** This is the time it takes the Web Interface to get the list of reconnections for this user from the XML service.	Secs	Compare the value of this measure with that of other *client start-up* metrics for a user to know what is the actual cause for the client start-up delay.
Session start-up server duration	This is the high-level server-side connection start-up metric. It includes the time spent on this virtual desktop to perform the entire start-up operation.	Secs	When this metric is high, it indicates that there is a server-side issue increasing session start times. To zero-in on this issue, compare the values of the *server start-up metrics* such as *Session creation server duration*, *Credentials obtention server duration, Program neighbourhood credentials obtention server duration, Credentials obtention network server duration, Credentials authentication server duration, Profile load server duration, Login script execution server duration , Drive mapping server duration, Drive mapping server duration,* and *Printer creation server duration*
Session creation server duration	Indicates the time spent by this virtual desktop in creating the session for this user.	Secs	This duration starts when the ICA client connection has been opened and ends when authentication begins. **This should not be confused with 'Session start-up server duration'.**

Measurement	Description	Measurement Unit	Interpretation
Credentials obtention server duration	Indicates the time taken by this virtual desktop to obtain the credentials of this user.	Secs	This time is only likely to be a significant if manual login is being used and the server-side credentials dialog is displayed (or if a legal notice is displayed before login commences). Because this metric may be artificially inflated if a user fails to provide credentials in a timely manner, it is not included in the *Session start-up server duration*. However, in the event that the user manually inputs the credentials, and the value of this measure is higher than that of all the other *client start-up* metrics that this test reports, it is a clear indicator that any connection delay that the user may have experienced is owing to slowness in obtaining user credentials.
Credentials obtentions network server duration	Indicates the time spent by this virtual desktop performing network operations to obtain credentials for this user.	Secs	This only applies to a Security Support Provider Interface login (a form of pass-through authentication where the client device is a member of the same domain as the server and Kerberos tickets are passed in place of manually entered credentials).
Program neighbourhood credentials obtention server duration	Indicates the time needed for this virtual desktop to cause the Program Neighborhood instance running on the client ("Program Neighborhood Classic") to obtain this user's credentials.	Secs	As in the case of the *Credentials obtention server duration* metric, because this metric may be artificially inflated if a user fails to provide credentials in a timely manner, it is not included in the *Session start-up server duration*.
Credentials authentication server duration	Indicates the time spent by this virtual desktop when authenticating the user's	Secs	Where server-side issues are causing user experience to deteriorate, you can compare the value of this measure

Measurement	Description	Measurement Unit	Interpretation
	credentials against the authentication provider, which may be Kerberos, Active Directory® or a Security Support Provider Interface (SSPI).		with that of all the other *server start-up metrics* that this test reports – i.e., *Session creation server duration* , *Credentials obtention server duration, Program neighbourhood credentials obtention server duration, Credentials obtention network server duration, Profile load server duration, Login script execution server duration, Drive mapping server duration, Drive mapping server duration,* and *Printer creation server duration* – to know what is the root-cause of delays in server start-up.
Profile load server duration	Indicates the time required by this virtual desktop to load this user's profile.	Secs	If this metric is high, consider your Terminal Services profile configuration. Citrix Consulting has found that when customers have logon times greater than 20 seconds, in most cases, this can be attributed to poor profile and policy design. Roaming profile size and location contribute to slow session starts. When a user logs onto a session where Terminal Services roaming profiles and home folders are enabled, the roaming profile contents and access to that folder are mapped during logon, which takes additional resources. In some cases, this can consume significant amounts of the CPU usage. Consider using the Terminal Services home folders with redirected personal folders to mitigate this problem. In general, consider using **Citrix Profile management** to manage user profiles in Citrix environments. This tool also provides logging capabilities to help

Measurement	Description	Measurement Unit	Interpretation
			isolate profile issues. If you are using Citrix profile management and have slow logon times, check to see if your antivirus software is blocking the Citrix profile management tool.
Login script execution server duration	Indicates the time needed by this virtual desktop to run this user's login script(s).	Secs	If the value of this measure is abnormally high for any user, consider if you can streamline this user or group's login scripts. Also, consider if you can optimize any application compatibility scripts or use environment variables instead.
Drive mapping server duration	Indicates the time needed for this virtual desktop to map this user's client drives, devices and ports.	Secs	Make sure that, when possible, your base policies include settings to disable unused virtual channels, such as audio or COM port mapping, to optimize the ICA protocol and improve overall session performance.
Printer creation server duration	Indicates the time required for this virtual desktop to synchronously map this user's client printers.	Secs	If the configuration is set such that printer creation is performed asynchronously, no value is recorded for this measure as it is does not impact completion of the session start-up. On the other hand, if excessive time is spent mapping printers, it is often the result of the printer autocreation policy settings. The number of printers added locally on the users' client devices and your printing configuration can directly affect your session start times. When a session starts, the virtual desktop has to create every locally mapped printer on the client device. Consider reconfiguring your printing policies to

Measurement	Description	Measurement Unit	Interpretation
			reduce the number of printers that get created - especially if users have a lot of local printers.

4.2.9 Virtual Desktop Sessions Details Test

A user logged into a virtual desktop does not imply active usage of that desktop. In a VDI infrastructure, it is common for users to just log into desktops, and leave them unused for long time periods. Such desktops are a huge resource drain, as they continue to consume resources, regardless of the level of activity on them. Idle users themselves are unproductive resources. Besides, since these users unnecessarily hold on to desktops, users with genuine needs may not have any desktops to work with. If administrators can quickly identify these idle users and the desktops they are logged into, they can rapidly pull the desktops from such users and assign them to users who can use them effectively. The **Virtual Desktop Sessions Details** test turns the spotlight on these idle users. For each user session on a virtual desktop, this test reports the total duration of the session and the percentage of time for which the session was active. The test also reports the total idle time during the session. From these statistics, administrators can accurately identify those users who are wasting the desktops assigned and resources allocated to them.

-

Target of the test : A Microsoft Hyper-V server

Agent deploying the test : An internal/remote agent

Outputs of the test : One set of results for every user who is currently logged into a virtual desktop

Configurable parameters for the test

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured.
3. **PORT** - The port at which the **HOST** listens. By default, this is NULL.
4. **INSIDE VIEW USING** - By default, this test communicates with every VM remotely and extracts “inside view” metrics. Therefore, by default, the **INSIDE VIEW USING** flag is set to **Remote connection to VM (Windows)**.

   Typically, to establish this remote connection, eG Enterprise requires that the eG agent be configured with domain administrator privileges. In high-security environments, where the IT staff might have reservations about exposing the credentials of their domain administrators, this approach to extracting “inside view” metrics might not be preferred. In such environments therefore, eG Enterprise provides administrators the option to deploy a piece of software called the **eG VM Agent** on every Windows

VM; this VM agent allows the eG agent to collect "inside view" metrics from the Windows VMs **without domain administrator rights.** Refer to Section 1.3.2 for more details on the **eG VM Agent**. To ensure that the "inside view" of Windows VMs is obtained using the eG VM Agent, set the **INSIDE VIEW USING** flag to **eG VM Agent (Windows)**. Once this is done, you can set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.

5. **DOMAIN, ADMIN USER, ADMIN PASSWORD,** and **CONFIRM PASSWORD** – By default, this test connects to each virtual guest remotely and attempts to collect "inside view" metrics. Accordingly, the **INSIDE VIEW USING** flag is set to **Remote connection to VM (Windows)** by default. To obtain a remote connection, the test must be configured with the privileges of an administrative user to the domain within which the guests reside. The first step towards this is to specify the **DOMAIN** within which the virtual guests reside. The **ADMIN USER** and **ADMIN PASSWORD** will change according to the **DOMAIN** specification. Discussed below are the different values that the **DOMAIN** parameter can take, and how they impact the **ADMIN USER** and **ADMIN PASSWORD** specifications:

   - **If the VMs belong to a single domain:** If the guests belong to a specific domain, then specify the name of that domain against the domain parameter. In this case, any administrative user in that domain will have remote access to all the virtual guests. Therefore, an administrator account in the given domain can be provided in the **ADMIN USER** field and the corresponding password in the **ADMIN PASSWORD** field. Confirm the password by retyping it in the **CONFIRM PASSWORD** text box.
   - I**f the VMs belong to different domains:**In this case, you might want to provide multiple domain names. If this is done, then, to access the guests in every configured domain, the test should be configured with the required user privileges; this implies that along with multiple **DOMAIN** names, multiple **ADMIN USER** names and **ADMIN PASSWORDS** would also have to be provided. To help administrators provide these user details quickly and easily, the eG administrative interface embeds a special configuration page. To access this page, simply click on the encircled '+' icon that appears alongside the . To know how to use the special page, refer to Section 3.6.1 of this document.
   - **If the INSIDE VIEW USING flag is set to 'eG VM Agent (Windows)'**: On the other hand, if the **INSIDE VIEW USING** flag is set to **eG VM Agent (Windows)**, then it implies that the inside view can be obtained without domain administrator privileges. Therefore, set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.

4. **REPORT BY USER** – For the *Hyper-V* monitoring model, the **REPORT BY USER** flag is set to **NO** by default, indicating that by default, the guest operating systems on the Hyper-V server are identified using the hostname specified in the operating system. On the other hand, for the *Hyper-V VDI* model, this flag is set to **YES** by default; this implies that in case of VDI servers, by default, the guests will be identified using the login of the user who is accessing the guest OS. In other words, in VDI environments, this test will, by default, report measures for every *username_on_virtualmachinename*.

5. **REPORT POWERED OS** - **This flag becomes relevant only if the REPORT BY USER flag is set to 'Yes'.**

   If the **REPORT POWERED OS** flag is set to **Yes** (which is the default setting), then this test will report measures for even those VMs that do not have any users logged in currently. Such guests will be identified by their *virtualmachine name* and not by the *username_on_virtualmachinename*. On the other hand, if the **REPORT POWERED OS** flag is set to **No**, then this test will not report measures for those

VMs to which no users are logged in currently.

6. **IGNORE VMS INSIDE VIEW** - Administrators of some high security Hyper-V environments might not have permissions to internally monitor one/more VMs. The eG agent can be configured to **not obtain the 'inside view' of such 'inaccessible' VMs** using the **IGNORE VMS INSIDE VIEW** parameter. Against this parameter, you can provide a comma-separated list of VM names, or VM name patterns, for which the inside view need not be obtained. For instance, your **IGNORE VMS INSIDE VIEW** specification can be: **xp,*lin*,win*,vista*. Here, the * (asterisk) is used to denote leading and trailing spaces (as the case may be). By default, this parameter is set to *none* indicating that the eG agent obtains the inside view of all VMs on a Hyper-V host by default.

   Note:
   While performing VM discovery, the eG agent will not discover the operating system of the VMs configured in the **IGNORE VMS INSIDE VIEW** text box.

7. **EXCLUDE VMS** - Administrators of some virtualized environments may not want to monitor some of their less-critical VMs - for instance, VM templates - both from 'outside' and from 'inside'. The eG agent in this case can be configured to completely exclude such VMs from its monitoring purview. To achieve this, provide a comma-separated list of VMs to be excluded from monitoring in the **EXCLUDE VMS** text box. Instead of VMs, VM name patterns can also be provided here in a comma-separated list. For example, your **EXCLUDE VMS** specification can be: **xp,*lin*,win*,vista*. Here, the * (asterisk) is used to denote leading and trailing spaces (as the case may be). By default, this parameter is set to *none* indicating that the eG agent obtains the inside and outside views of all VMs on a virtual host by default. By providing a comma-separated list of VMs/VM name patterns in the **EXCLUDE VMS** text box, you can make sure the eG agent stops collecting 'inside' and 'outside' view metrics for a configured set of VMs.

8. **IGNORE WINNT** – By default, the eG agent does not support the *inside view* for VMs executing on **Windows NT** operating systems. Accordingly, the **IGNORE WINNT** flag is set to **Yes** by default.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Total time in session:	Indicates the time that has elapsed since this user logged into this desktop.	Mins	
Active time in last measure period:	Indicates the percentage of time in the last measurement period during which this user actively used this desktop.	Percent	Ideally, the value of this measure should be 100%. A low value for this measure denotes a high level of inactivity recently.
Time since last activity:	Indicates the time that has elapsed since this user	Mins	A high value for this measure indicates that the user has been idle for a long

Measurement	Description	Measurement Unit	Interpretation
	performed an action on this desktop.		time. Compare the value of this measure across users to know which user has been idle for the longest time.
Total idle time in session:	Indicates the total time for which this user was idle during the session.	Mins	If the value of this measure is the same as the value of the *Total time in session* measure for a user, it means that the user has been idle throughout the session. If the value of this measure is close to the value of the *Total time in session* measure for a user, it implies that the user has been idle for a long time. If the value of this measure is much lesser than the value of the *Total time in session* measure for a user, it means that the user has been active for most part of the session.

Conclusion

This document has described in detail the monitoring paradigm used and the measurement capabilities of the eG Enterprise suite of products with respect to **Microsoft Hyper-V environments**. For details of how to administer and use the eG Enterprise suite of products, refer to the user manuals.

We will be adding new measurement capabilities into the future versions of the eG Enterprise suite. If you can identify new capabilities that you would like us to incorporate in the eG Enterprise suite of products, please contact support@eginnovations.com. We look forward to your support and cooperation. Any feedback regarding this manual or any other aspects of the eG Enterprise suite can be forwarded to feedback@eginnovations.com.