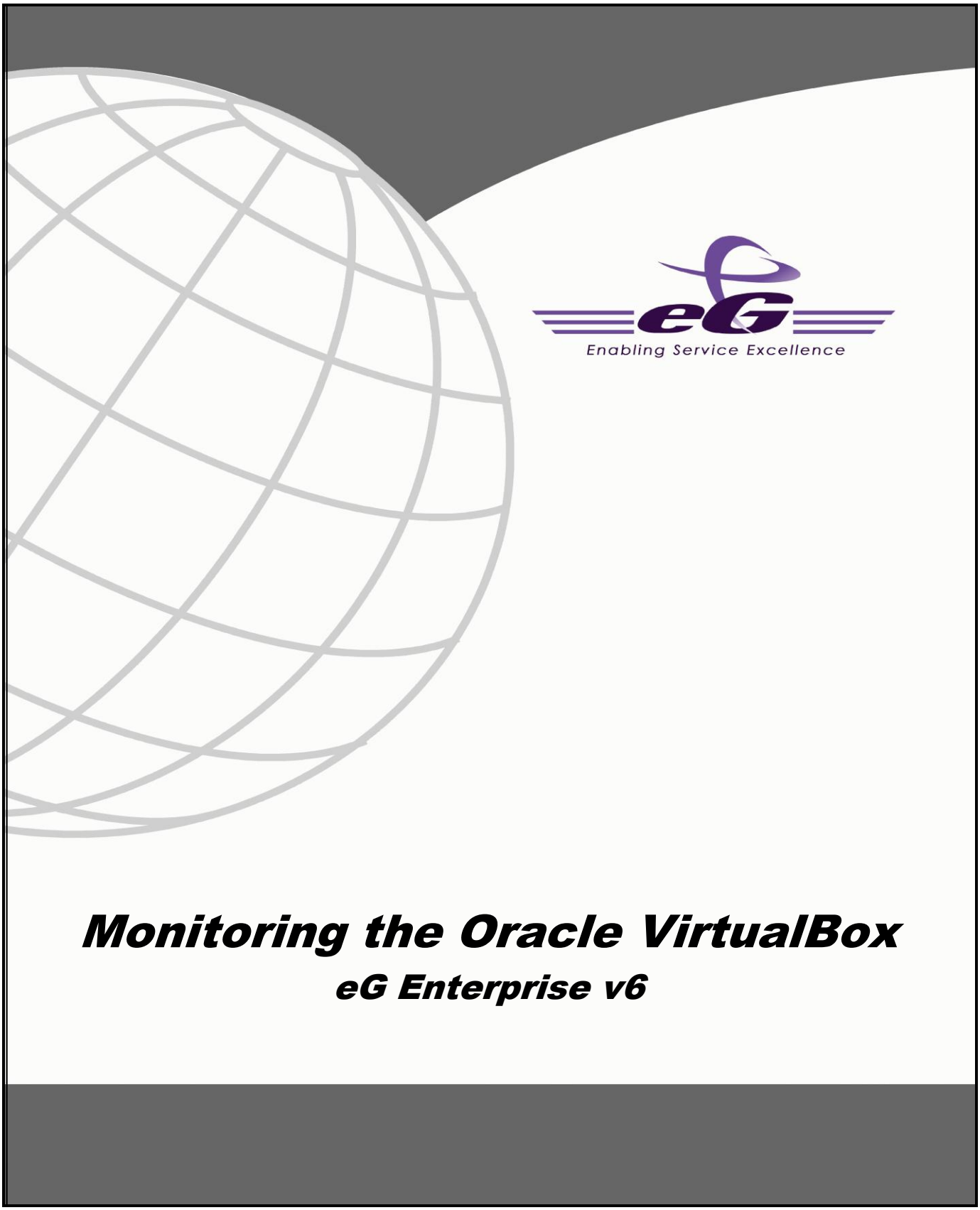


Monitoring the Oracle VirtualBox

eG Enterprise v6

Restricted Rights Legend

Trademarks

Copyright

Table of Contents

Table of Figures

Chapter 1

Introduction

Oracle VDI (Virtual Desktop Infrastructure) provides desktop virtualization to replace personal computers with virtual machines (VMs) on a server. Users can access these VMs though any RDP client, or through the web via Sun Secure Global Desktop (SGD).

Oracle Virtual Desktop Infrastructure is made up of four main components: virtualization platform, session management (Oracle VDI Core), desktop access clients, and storage.

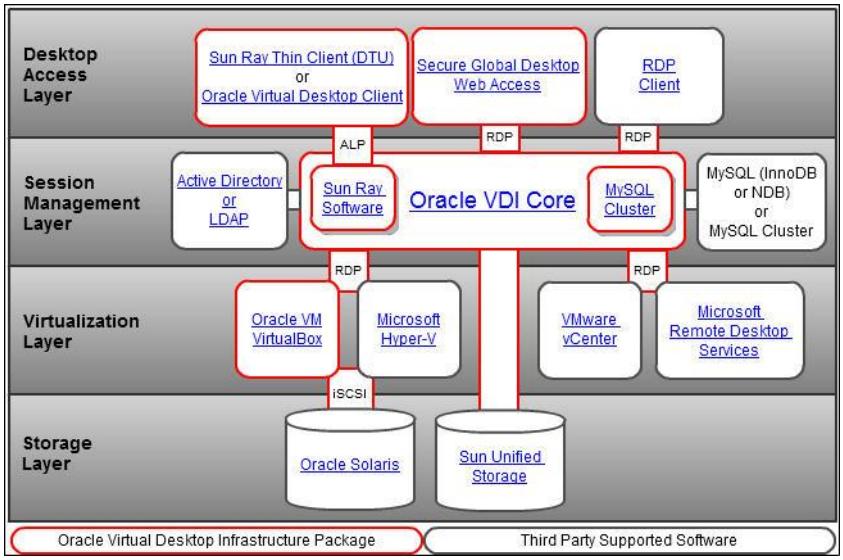


Figure 1.1: Architecture of the Oracle Virtual Desktop Infrastructure

The basis for the Oracle Virtual Desktop Infrastructure architecture is the virtualization platform. In addition to creating and storing virtual machines, the virtualization platform offers the core functionality needed for virtual desktop management such as starting, stopping, and snapshotting virtual machines. Oracle Virtual Desktop Infrastructure 3.2 supports Oracle VM VirtualBox (the Oracle VDI Hypervisor), VMware vCenter, Microsoft Hyper-V, and Microsoft Remote Desktop Services as virtualization platforms.

Oracle VM VirtualBox is cross-platform x86 virtualization software that extends the power of your existing computers to run multiple operating systems, on the same hardware, at the same time, and alongside your existing applications.

Oracle VM VirtualBox includes a hypervisor for the host platform, an application programming interface (API) and software development kit (SDK) for managing guest virtual machines, a command-line tool for managing guests locally, a web service for remote management of guests, a wizard-style

graphical tool to manage guests, a graphical console for displaying guest applications on the local host, and a built-in Remote Desktop Protocol (RDP) server that provides complete access to a guest from a remote client. At the core is the hypervisor, implemented as a *ring 0* (privileged) kernel service. Figure 1.2 shows the relationships between all of these components.

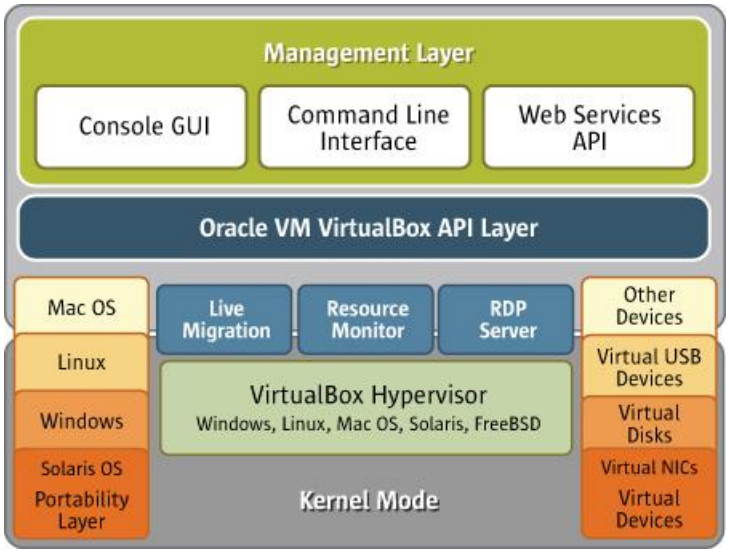


Figure 1.2: Architecture of an Oracle VirtualBox

The kernel service consists of a device driver named *vboxsrv*, which is responsible for tasks such as allocating physical memory for the guest virtual machine, and several loadable hypervisor modules for things like saving and restoring the guest process context when a host interrupt occurs, turning control over to the guest OS to begin execution, and deciding when VT-x or AMD-V events need to be handled.

The hypervisor does not get involved with the details of the guest operating system scheduling. Instead, those tasks are handled completely by the guest during its execution. The entire guest is run as a single process on the host system and will run only when scheduled by the host. If they are present, an administrator can use host resource controls such as scheduling classes and CPU caps or reservations to give very predictable execution of the guest machine.

Additional device drivers will be present to allow the guest machine access to other host resources such as disks, network controllers, and audio and USB devices. In reality, the hypervisor actually does little work. Rather, most of the interesting work in running the guest machine is done in the guest process. Thus the host's resource controls and scheduling methods can be used to control the guest machine behavior.

In addition to the kernel modules, several processes on the host are used to support running guests. All of these processes are started automatically when needed.

1.1 Challenges in Monitoring the Oracle VirtualBox

What makes monitoring an Oracle VDI infrastructure a challenge is the large number of virtual desktops that will typically be configured on the VirtualBoxes and the large number of users to the desktops. While it can be very difficult to keep track of which user is accessing which desktop on which VirtualBox, the live and automatic migration of desktops to other VirtualBoxes (if any) in the environment only compounds the problem. To make matters worse, the users to the VDI service demand from the virtual desktops the same quality of service that they are used to receiving from their physical desktops. This means, quick access, uninterrupted connectivity, and stable operations

will be the criteria on which the user experience with the VDI service will be judged. Non-availability of a desktop when a user needs it, or slowdowns in desktop operations caused by a resource contention at the desktop-level or at the host-level may result in a deluge of user complaints and a bevy of dissatisfied users. To avoid this, service desk should be able to:

- Continuously monitor the user activity to the virtual desktops operating on a VirtualBox;
- Know which user accessed the desktop on which VirtualBox at what time;
- Track the powered-on state of desktops;
- Study the resource usage patterns of the virtual desktops to nail the root-cause of resource contentions - is it owing to a resource-hungry desktop? or a resource-starved host?;
- Promptly alert users to potential resource drains on desktops or a sudden change in the desktop state, much before users notice the difference!

1.2 eG's Solution to Oracle VirtualBox Monitoring

The specialized *Oracle VirtualBox* monitoring model addresses all the requirements outlined above and more! This 100%, web-based solution employs a single eG agent to perform detailed 'In-N-Out' monitoring of the virtual desktops operating on an Oracle VirtualBox.

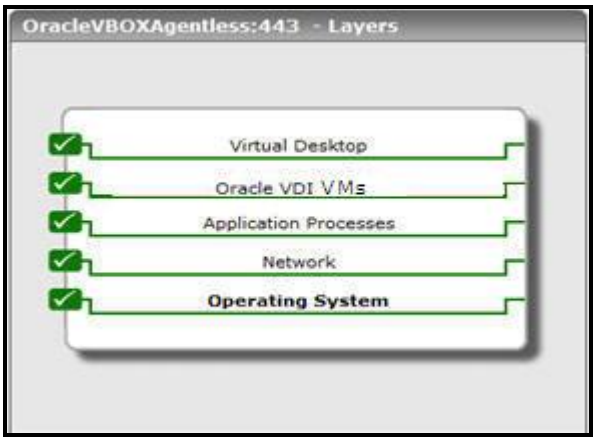


Figure 1.3: The layer model of Oracle VirtualBox

The metrics collected by this eG agent report on the percentage of the Oracle VirtualBox host's resources that each of the VMs on the server are using - i.e., the relative loading of the guest VMs. This represents the view of how a guest VM and its applications are doing - from the "outside" - i.e., from outside the guest VM.

In addition, the eG agent also connects to each guest VM that is currently powered on and determines the guest OS version, the name(s) of the users who are logged on, the applications they are accessing, and the resource usage of the applications running inside the guest (as seen from within the guest operating system). This represents the view from within the guest operating system - i.e., the "inside" view.

In addition, the same agent can also track the critical processes running on the Oracle VirtualBox host and their resource usage, the network connection to the host, and the TCP connectivity of the host, and thus report on the overall health of the host.

The agent is also capable of capturing and reporting on the live migration of desktops from one VirtualBox to another.

Based on the monitoring approach chosen, you can deploy this eG agent on the VirtualBox itself or on any remote Windows/Linux/Solaris host in the environment. Section 1.3 discusses both these deployment models in detail.

1.3 Agent Deployment Models

eG Enterprise allows administrators the flexibility to choose between the *agent-based* and *agentless* approaches to monitoring the *Oracle VirtualBox*.

1.3.1 The Agent-based Monitoring Approach

The agent-based approach **requires that the eG agent be installed on the host operating system of the VirtualBox.** Since Oracle VDI comes bundled with an Oracle VirtualBox that runs an Oracle Solaris operating system, you need to install a **Solaris eG agent** on the host. To know how to install an eG agent on Solaris, refer to the *eG Installation Guide*.

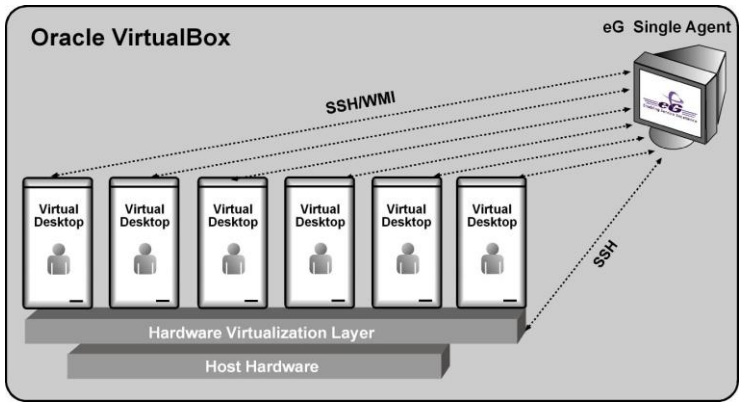


Figure 1.4: Agent-based monitoring of the Oracle VirtualBox

This agent should then be configured to communicate with the Oracle VirtualBox via SSH and run privileged Virtual Desktop Access (VDA) commands on the Oracle VirtualBox to determine the health of the host, to discover the IP address and operating system of each of the guests on the host, and to report how each guest has utilized the host's physical resources (i.e., *outside view*). For connecting to the target Oracle VirtualBox via SSH, the eG agent has to be configured with the credentials of a user with the required privileges. Also, to enable the eG agent to run VDA commands on the host, a **sudo** package has to be installed on the VirtualBox. To know how, refer to Section 1.4 below.

Once the guests are discovered, the eG agent remotely communicates with each guest using SSH/WMI (depending upon the operating system of the guest) to obtain the "inside view" of every guest. To establish this remote connection with Windows VMs in particular, 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 on the service console to collect "inside view" metrics from the VMs **without domain administrator rights.** Refer to Section 1.5 for more details on the **eG VM Agent**.

1.3.2 The Agentless Monitoring Approach

The **agentless approach** to monitoring the Oracle VirtualBox involves the following:

- Deploying the eG agent on a remote system running Microsoft Windows or Linux or Solaris;
- Configuring the remote eG agent to communicate with the target VirtualBox via SSH;
- Configuring the remote eG agent to run Virtual Desktop Access (VDA) commands on the VirtualBox to perform guest discovery and to collect host-level and 'outside view' metrics; to run these commands, you need to install a **sudo** package on the VirtualBox - refer to Section 1.4 to know how.
- Configuring the remote eG agent to collect performance metrics from each of the guest VMs configured on the VirtualBox using SSH/WMI; by default, the eG agent uses SSH/WMI (depending upon the virtual OS to be monitored) to communicate remotely with the virtual machines on the VirtualBox (see Figure 1.5) and collect metrics. To establish this remote connection with Windows VMs in particular, 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 1.5 for more details on the **eG VM Agent**.

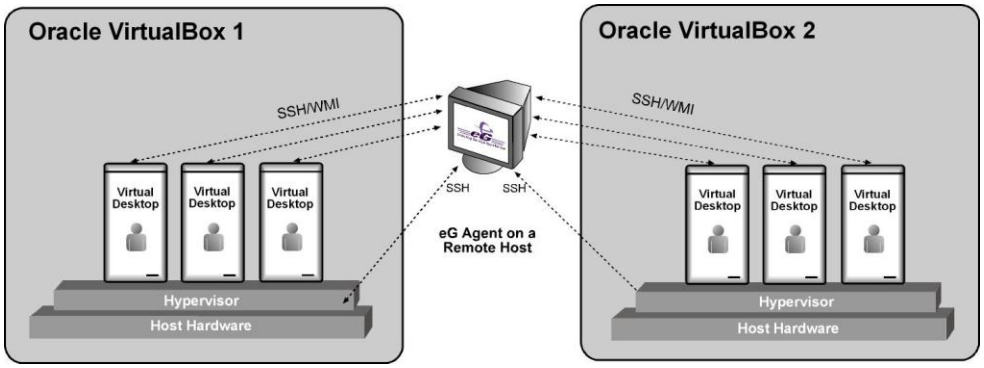


Figure 1.5: The Agentless approach to monitoring the Oracle VirtualBox

Regardless of the monitoring approach chosen, certain pre-requisites need to be fulfilled before attempting to monitor the Oracle VirtualBox. Section 1.4 below discusses these pre-requisites.

1.4 Pre-requisites for Monitoring the Oracle VirtualBox

This section details the pre-requisites that need to be fulfilled for monitoring an Oracle VirtualBox in an agent-based and an agentless manner.

1.4.1 Pre-requisites for Monitoring the Oracle VirtualBox in an Agent-based Manner

1.4.1.1 General Pre-requisites

Enable the eG agent to communicate with the eG manager port (default: 7077).

1.4.1.2 Pre-requisites for Auto-Discovering VMs and Obtaining their "Outside View"

- Ensure that the eG agent is able to connect to the target VirtualBox via SSH.
- Make sure that the SSH port (default: 22) is opened for communication between the eG agent and the Oracle VirtualBox.
- Configure all the tests that the eG agent executes with the name and password of a user who has the right to access the target Oracle VirtualBox via SSH.
- To enable the eG agent to run VDA commands on the VirtualBox, install the **sudo** package on the VirtualBox. To install this package, do the following:
    - Login to the Solaris system hosting the VirtualBox as a *root* user.
    - To download the **sudo** package, connect to the URL: http://sysinfo.bascomp.org/solaris/installing-sudo-on-solaris/
    - If the Solaris processor is Intel based, download the file **TCMsudo-1.8.2-i386.pkg.gz** from the web site mentioned above. On the other hand, if the Solaris host uses a SPARC processor instead, download the file **TCMsudo-1.8.2-sparc.pkg.gz** from the web site.
    - Download the chosen file to any location on the VirtualBox host (say, **/tmp**).
    - From the Solaris prompt, switch to the directory hosting the downloaded package and unzip the compressed package using the following command:

      gunzip <package_name>

      For instance:

      gunzip TCMsudo-1.8.2-sparc.pkg.gz
    - Then, install the package by issuing the following command at the prompt:

      pkgadd -d <package name>

      For instance:

      pkgadd –d TCMsudo-1.8.2-sparc.pkg
    - Once installation is complete, you will find that the package is installed in the **/usr/local/** folder on the Solaris host.
- All the tests run by the eG agent should be configured with the full path to the install directory of the **sudo** package;

1.4.1.3 Pre-requisites for Obtaining the "Inside View" of Windows VMs, using the eG VM Agent

- Install the eG VM Agent on each Windows VM. For details on how to install the eG VM Agent, refer to Section 1.5 of this document.
- Enable the eG agent to communicate with the port at which the eG VM Agent listens (default port: 60001).

- Set the **INSIDE VIEW USING** flag for all the "inside view" tests to **eG VM Agent (Windows)**.

1.4.1.4 Pre-requisites for Obtaining the "Inside View" of VMs, without using the eG VM Agent

- Ensure that the eG agent has IP connectivity to at least one of the network interfaces of the VMs.
- Typically, the Windows File and Print Sharing port is 139. Enable the eG agent to communicate with this port.
- The **ADMIN$** share should be enabled for all Windows-based virtual guests being monitored and the administrative account must have permissions to this share drive. Refer to Section 1.6.1 of this document for a step-by-step procedure to achieve this.
- To enable the eG agent to communicate with the Windows VMs, an administrative account login and password (either a local account or a domain account) must be provided when configuring the eG monitoring capabilities.
- In case of VMs with the Windows XP/Windows 2003/Windows 2008/Windows Vista/Windows 7 operating systems, the firewall on the guest should be explicitly configured to allow Windows File and Print Sharing services which are required for the agent to communicate with the guest operating system. Refer to Section 1.6.2 of this document for a step-by-step procedure to achieve this.
- Set the **INSIDE VIEW USING** flag for all the "inside view" tests to **Remote connection to VM (Windows)**.
- For monitoring a Linux/Solaris VM, the SSH port (TCP port 22) must be enabled for communication between the eG agent and the VM being monitored.

1.4.2 Pre-requisites for Monitoring the Oracle VirtualBox in an Agentless Manner

1.4.2.1 General Pre-requisites

- Enable the remote agent to communicate with the eG manager port (default: 7077).
- If VMs running on multi-byte operating systems are to be monitored (eg., *Windows Japanese*), then the remote agent monitoring such VMs should also run on a multi-byte operating system.

1.4.2.2 Pre-requisites for Auto-Discovering VMs and Obtaining their "Outside View"

- Ensure that the remote agent has IP connectivity to the target VirtualBox.
- Ensure that the remote agent can connect to the target VirtualBox via SSH.
- Configure all the tests that the remote agent executes with the name and password of a user who is privileged to access the VirtualBox via SSH.
- To enable the remote agent to run VDA commands on the VirtualBox, a **sudo** package has to be installed on the VirtualBox host; to know how to install the **sudo** package, refer to Section 1.4.1.2 above.
- After the **sudo** package is installed, perform the following steps on the VirtualBox host:
    - Login to the host as a *root* user;

- At the command prompt of the host, issue the following command to create a new user:

  useradd –d /export/home/<username> –m <username>

  For instance:

  useradd –d /export/home/eguser –m eguser

- Next, issue the following command to set a password for the above user:

  passwd <username>

- When prompted to provide the password, specify the same.

- Then, proceed to edit the **sudo** script by issuing the following command:

  usr/local/sbin/visudo

- Add the following entry to the script:

  <username> ALL=NOPASSWD:/usr/bin/VBoxManage

- All the tests run by the eG agent should be configured with the full path to the install directory of the **sudo** package;

1.4.2.3 Pre-requisites for Obtaining the "Inside View" of Windows VMs, using the eG VM Agent

- Install the eG VM Agent on each Windows VM. For details on how to install the eG VM Agent, refer to Section 1.5 of this document.
- Enable the remote agent to communicate with the port at which the eG VM Agent listens (default port: 60001).
- Set the **INSIDE VIEW USING** flag for all the "inside view" tests to **eG VM Agent (Windows)**.

1.4.2.4 Pre-requisites for Obtaining the "Inside View" of VMs, without using the eG VM Agent

- Ensure that the remote agent has IP connectivity to at least one of the network interfaces of the VMs.
- The **ADMIN$** share should be enabled for all Windows-based virtual guests being monitored and the administrative account must have permissions to this share drive. Refer to Section 1.6.1 of this document for a step-by-step procedure to achieve this.
- To enable the remote agent to communicate with the Windows VMs, an administrative account login and password (either a local account or a domain account) must be provided when configuring the eG monitoring capabilities.
- In case of VMs with the Windows XP/Windows 2003/Windows 2008/Windows Vista/Windows 7 operating systems, the firewall on the guest should be explicitly configured to allow Windows File and Print Sharing services which are required for the remote agent on the vSphere/ESX host to communicate with the guest operating system. Refer to Section 1.6.2 of this document for a detailed procedure.

- For monitoring a Windows VM, TCP port 139 must be accessible from the remote agent to the VM.
- For monitoring a Linux/Solaris VM, the SSH port (TCP port 22) must be enabled for communication between the remote agent and the VM being monitored.
- For obtaining the "inside view" of VMs running Windows Vista/Windows 7/Windows 2008 operating systems, the **eGurkhaAgent** service of the eG remote agent should be configured to run using *domain administrator* privileges. Refer to the *eG User Manual* for the procedure. For obtaining the "inside view" of other Windows VMs however, the remote agent service requires no such privileges.
- Set the **INSIDE VIEW USING** flag for all the "inside view" tests to **Remote connection to VM (Windows)**.

1.5 Configuring the eG Agent to Obtain the Inside View of Windows VMs, using the eG VM Agent

To provide the inside view of a Unix VM, the eG agent uses secure shell (SSH). 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**.

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 includes;
- 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;

Use the install procedure that is ideal for your environment, and quickly get the eG VM Agent up and running. 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 1.6 then appears. Click on the **Next** button in Figure 1.6 to continue.

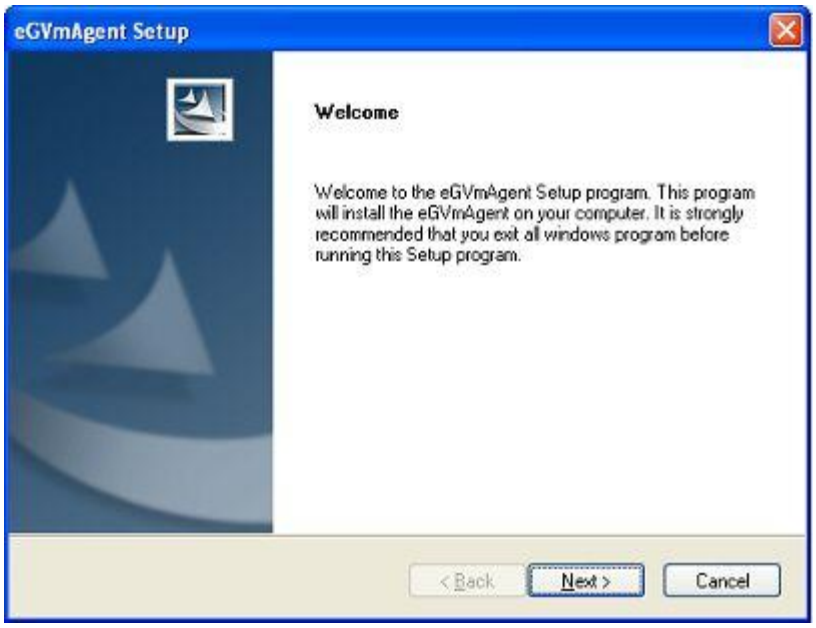


Figure 1.6: Welcome screen of the eG VM Agent installation wizard

3. When Figure 1.7 appears, click on **Yes** to accept the displayed license agreement.

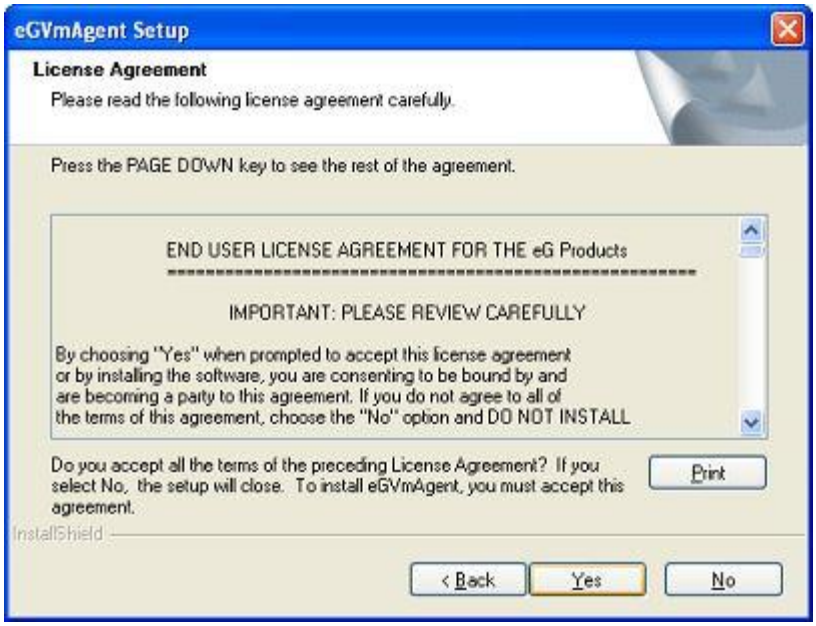


Figure 1.7: Accepting the license agreement

4. Use the **Browse** button in Figure 1.8 to indicate the location in which the agent should be installed, and click the **Next** button to proceed.

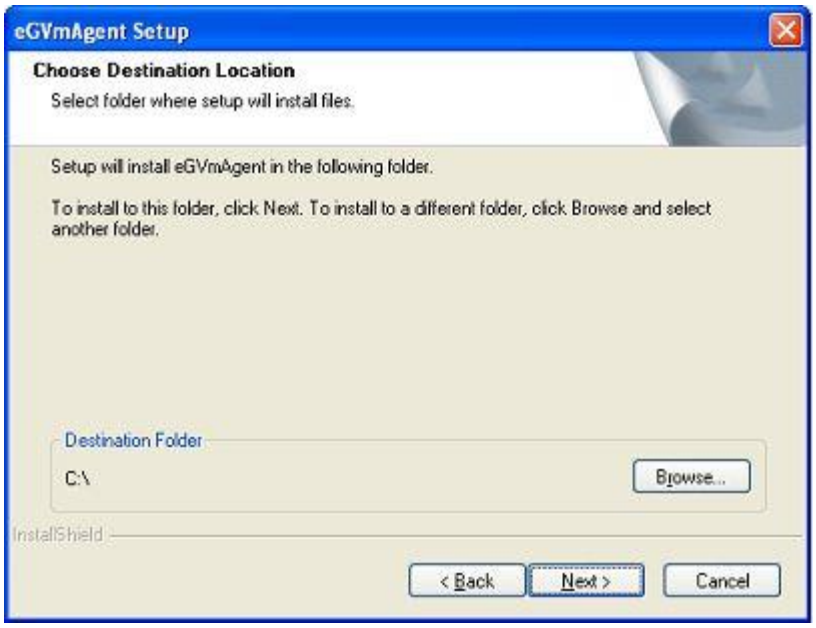


Figure 1.8: Specifying the install directory of the eG VM Agent

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 1.9 to proceed.

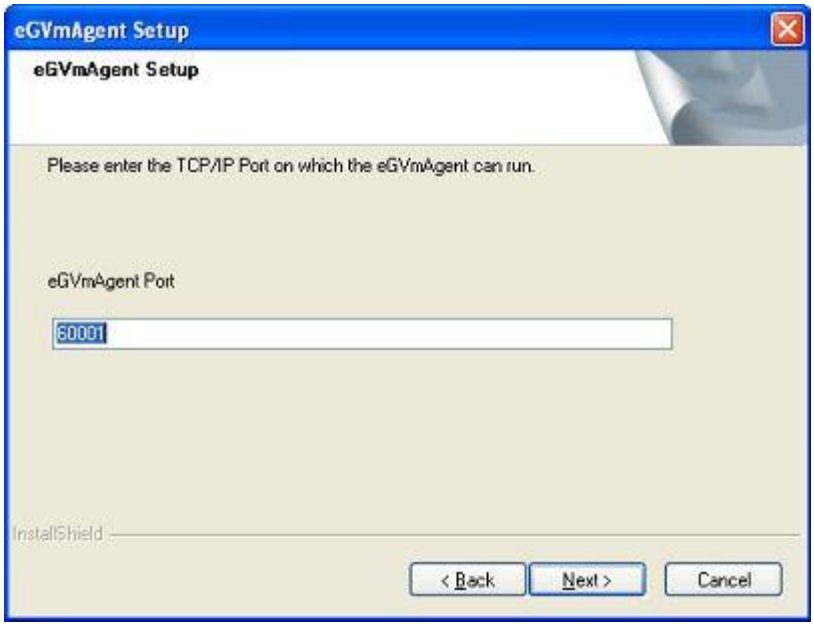


Figure 1.9: Specifying the VM agent port

6. A summary of your specifications then follows (see Figure 1.10). Click **Next** to proceed.

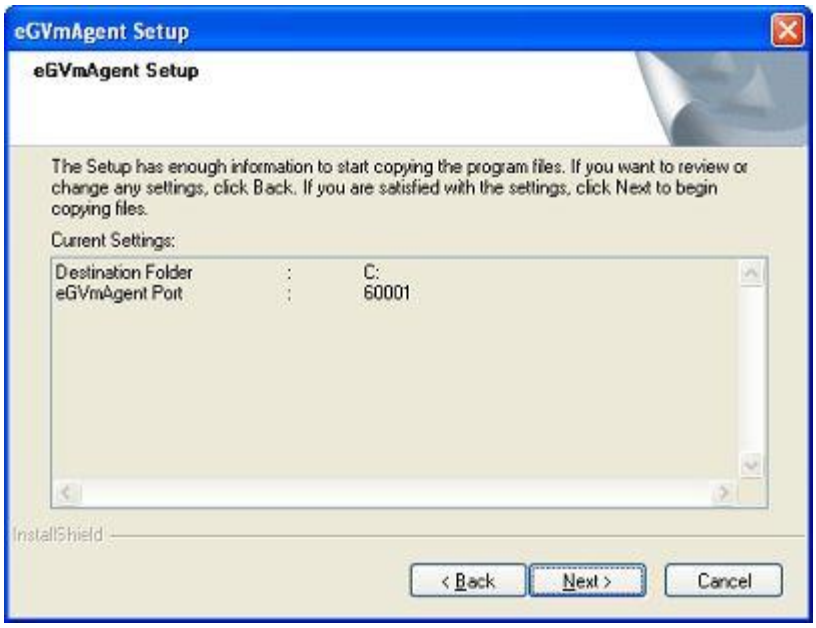


Figure 1.10: A summary of your specifications

7. Finally, click the **Finish** button in Figure 1.11 to complete the installation.

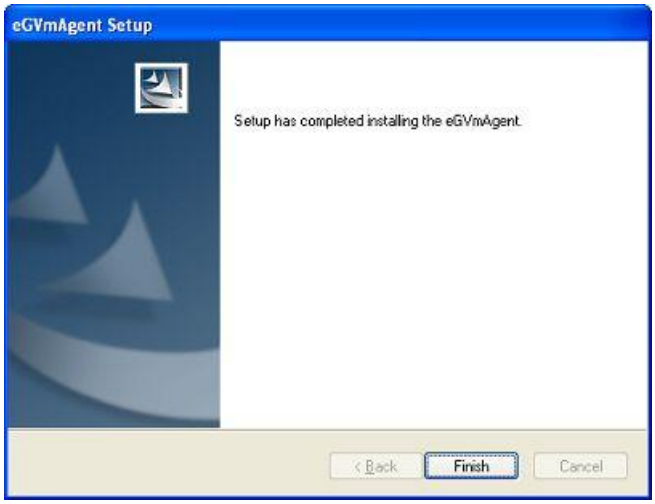


Figure 1.11: Finishing the installation

1.5.1 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 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.

1.5.2 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.

1.5.3 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 Enterprise Suite 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.

1.6 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 uses SSH/WMI (depending upon the virtual OS to be monitored) to communicate remotely with the virtual machines on VirtualBox and collect metrics. To establish this remote connection with Windows VMs in particular, eG Enterprise requires that the eG agent 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 following pre-requisites need to be fulfilled:

- The **ADMIN$** share will have to be available on the Windows guests
- The Windows Firewall should be configured to allow Windows File and Print Sharing

The sections to come discuss the procedure to be followed for fulfilling the 2 requirements above.

1.6.1 Enabling ADMIN$ Share Access on Windows Virtual Guests

1.6.1.1 Enabling ADMIN$ Share Access on Windows 2000/2003 VMs

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 Figure 1.12 appears indicating the same.

Figure 1.12: The ADMIN$ share does not exist

On the other hand, if the **ADMIN$** share pre-exists, Figure 1.13 appears. In such a case, first, remove the **ADMIN$** share by selecting the **Do not share this folder** option from Figure 1.13 and clicking the **Apply** and **OK** buttons. After this, you will have to repeat step 1 of this procedure to open Figure 1.12. Then, proceed as indicated by step 3 onwards.

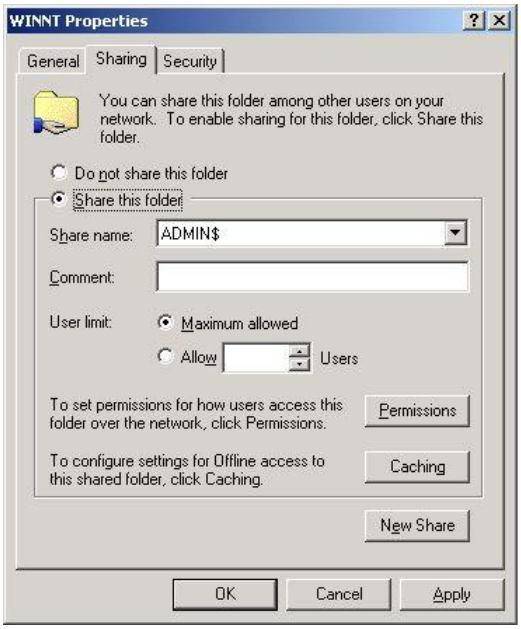


Figure 1.13: Admin$ share pre-exists

3. To create (or re-create) the **ADMIN$** share, select the **Share this folder** option from Figure 1.14, and provide **ADMIN$** share against the **Share name** text box (see Figure 1.14).

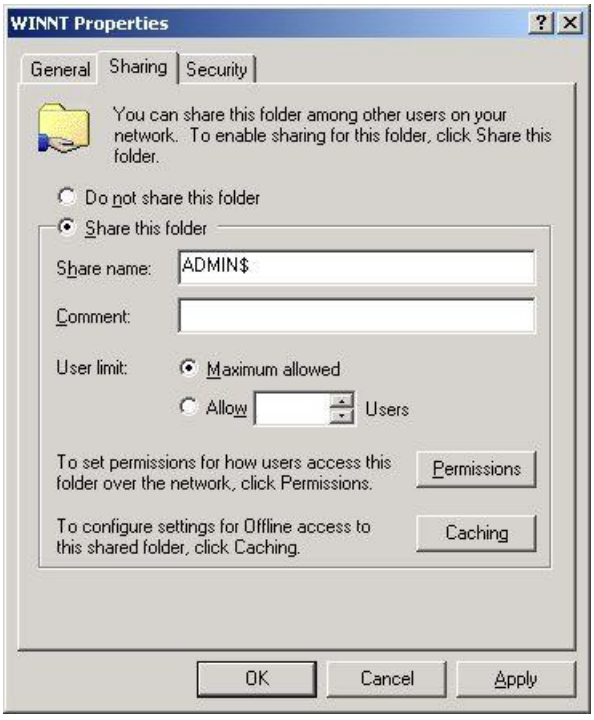


Figure 1.14: Creating the ADMIN$ share

4. 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 (local/domain); also, the **credentials of this user should be passed while configuring the eG monitoring**

capabilities - i.e., while configuring the VMware tests. To grant the access permissions, click on the **Permissions** button in Figure 1.14.

5. By default, the **ADMIN$** share can be accessed by **Everyone** (see Figure 1.15). To grant access rights to a specific administrative (local/domain) user, select the **Add** button in Figure 1.15. When Figure 1.16 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.

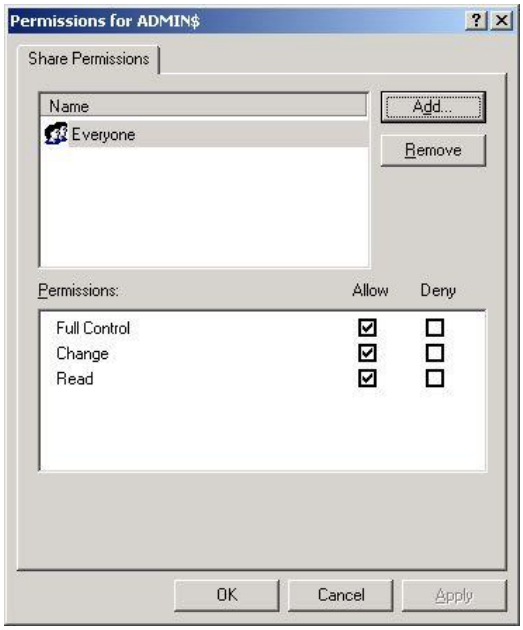


Figure 1.15: Clicking the Add button

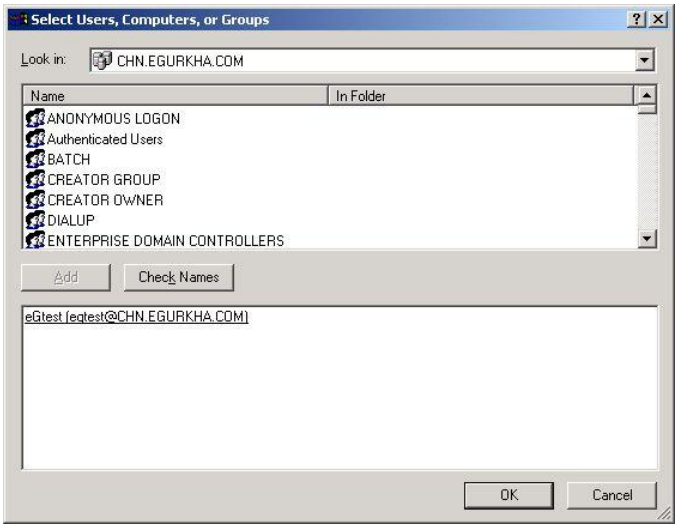


Figure 1.16: Selecting the administrative user to whom access rights are to be granted

6. Finally, click the **OK** button. You will then switch to Figure 1.17, where the newly added administrator account will appear.

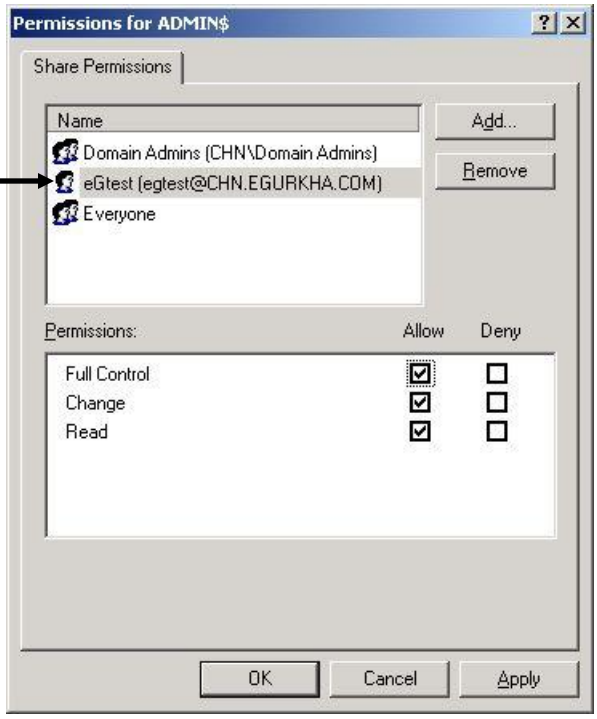


Figure 1.17: The administrator account granted access permissions

7. Select the newly added administrator account from Figure 1.17, and then, using the **Permissions** section, grant the administrator **Full Control**, **Change**, and **Read** permissions.
8. Finally, click the **Apply** and **OK** buttons in Figure 1.17 to register the changes.
9. Once you return to Figure 1.18, click on the **Security** tab to define the security settings for the **ADMIN$** share (see Figure 1.18).

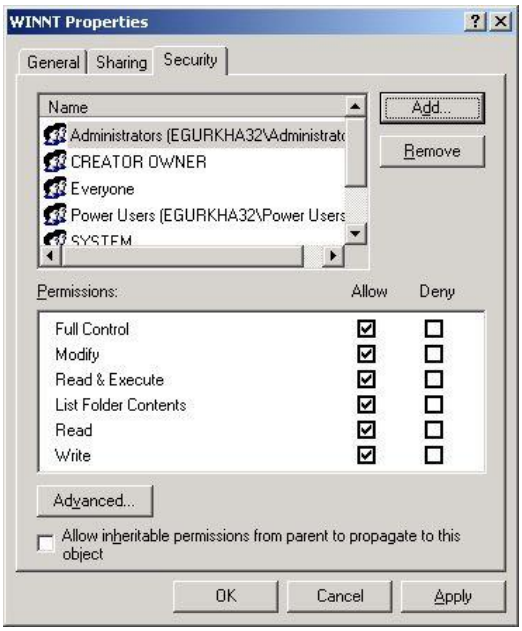


Figure 1.18: Defining the Security settings for the ADMIN$ share

10. 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 1.18, pick a domain from the **Look in** list of Figure 1.19, select the said administrator account from the domain users list below, and click the **Add** button (in Figure 1.19) to add the chosen account. Then, click the **OK** button in Figure 1.19.

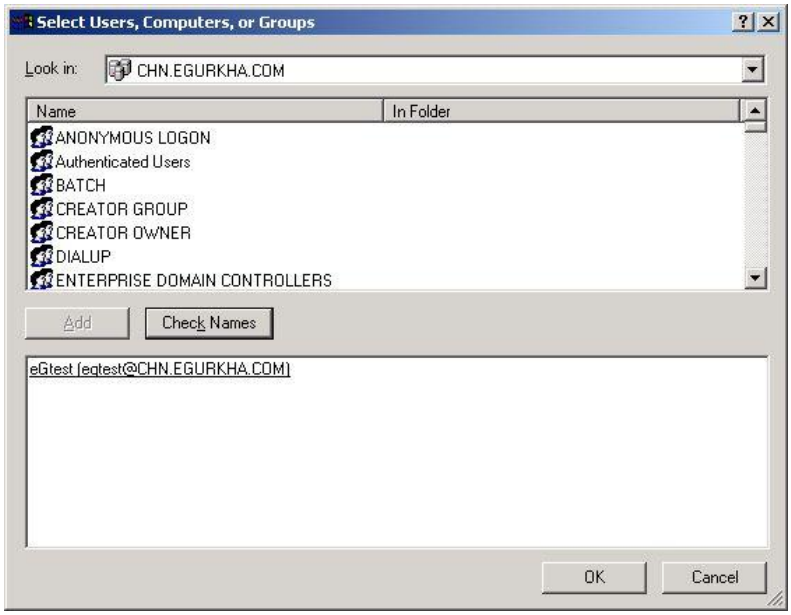


Figure 1.19: Adding the administrator account

11. This will bring you back to Figure 1.18, but this time, the newly added domain administrator account will be listed therein as indicated by Figure 1.20.

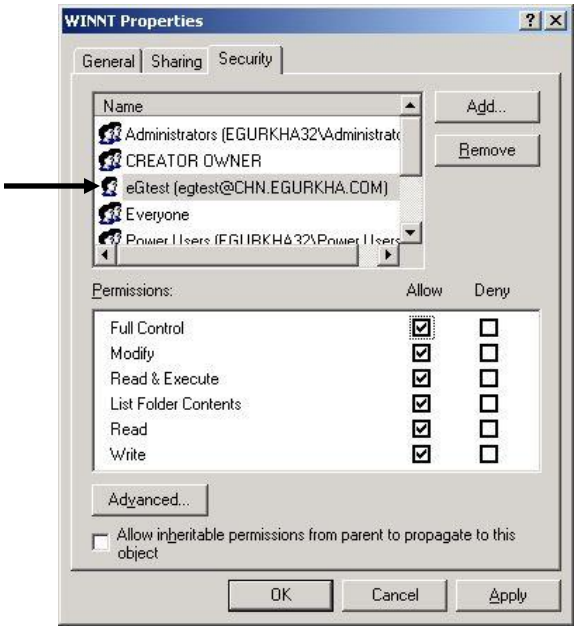


Figure 1.20: The Administrator account in the Security list

12. Finally, click the **Apply** and **OK** buttons in Figure 1.20.

1.6.1.2 Enabling ADMIN$ Share Access on Windows 2008 VMs

To enable the **ADMIN$** share on a Windows 2008 VM, do the following:

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 **Share** option from the shortcut menu.

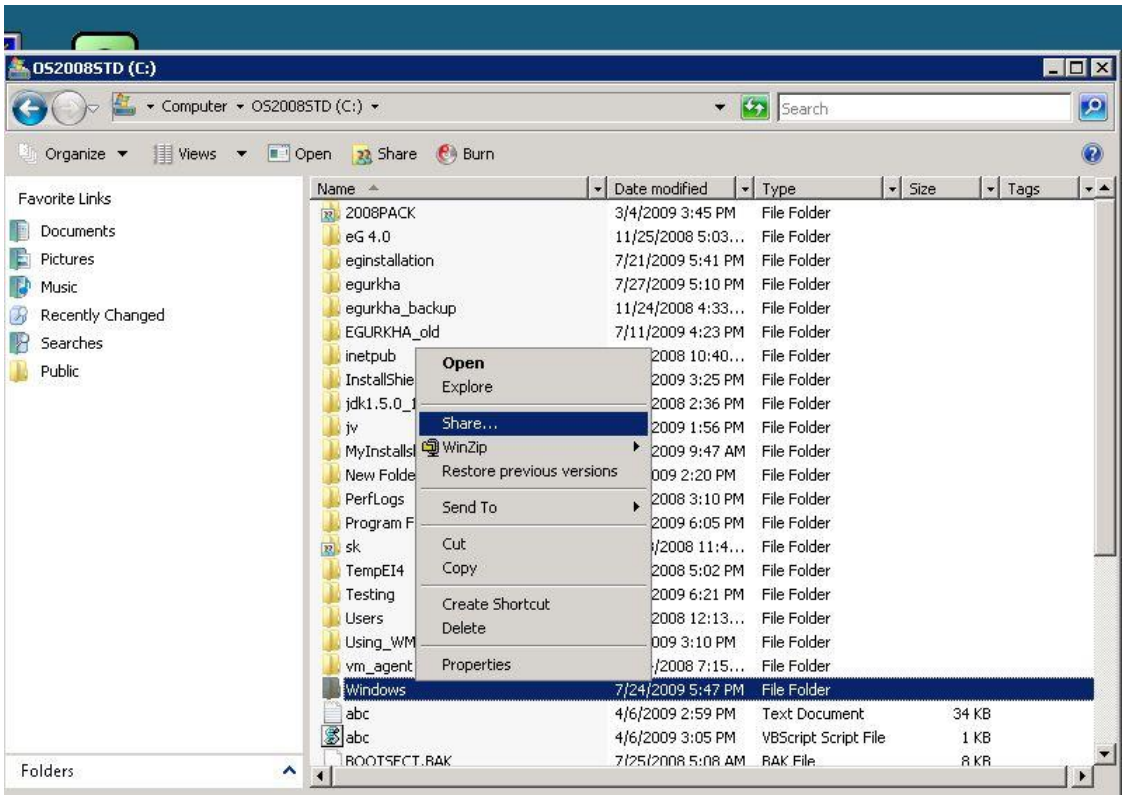


Figure 1.21: Selecting the Share option from the shortcut menu

2. Figure 1.22 will then appear. Click on **Advanced Sharing** in Figure 1.22.

Figure 1.22: Cicking on Advanced Sharing

3. Select the **Share this folder** check box in Figure 1.23 that appears, enter **ADMIN$** against **Share name**, and click on the **Permissions** button in Figure 1.23, to allow only a local/domain administrator to access the folder.

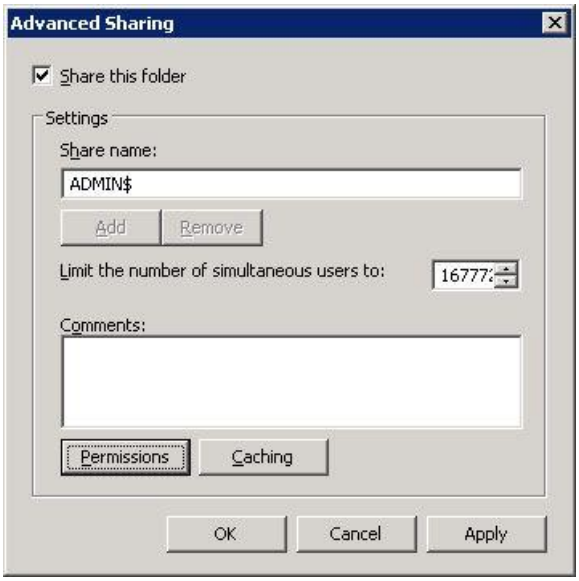


Figure 1.23: Enabling the ADMIN$ share

When Figure 1.24 appears, click on the **Add** button therein.

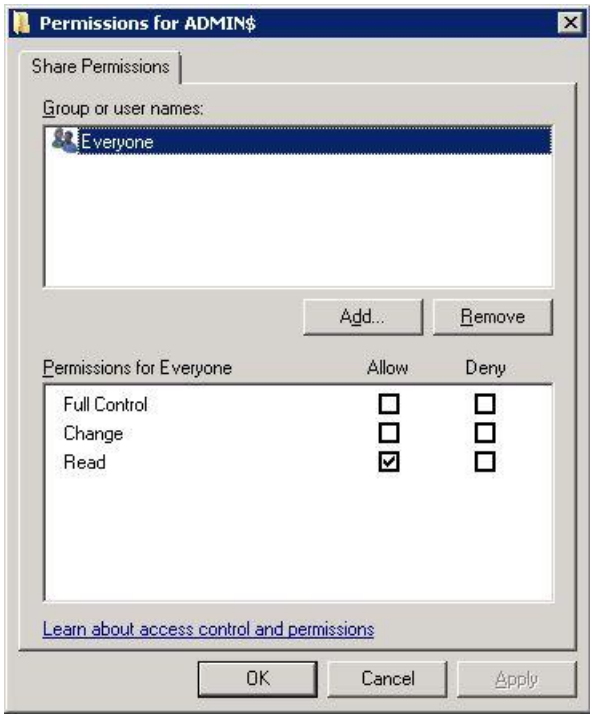


Figure 1.24: Clicking on the Add button

4. To allow a domain administrator to access the folder, first, ensure that a valid domain is specified in the **From this location** box of Figure 1.25. If you want to grant access to a local administrator instead, ensure that the name of the local host is displayed in the **From this location** box. To change this specification, use the **Locations** button in Figure 1.25. Then, enter the name of the local/domain administrator in the **Enter the object names to select** text area, and click the **OK** button.

Figure 1.25: Allowing a domain administrator to access the folder

5. The newly added user will be listed in the **Group or user names** section, as depicted by Figure 1.26. Select this user, and then, check all the three check boxes under **Allow** in the **Permissions for <user>** section in Figure 1.26. Then, click the **Apply** and **OK** buttons therein.

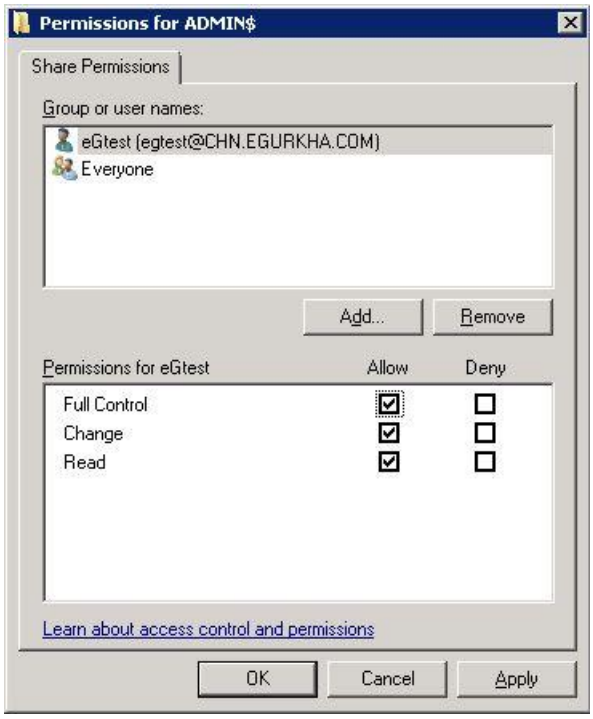


Figure 1.26: Allowing full access to the local/domain administrator

6. When Figure 1.27 appears, click on the **Apply** and **OK** buttons therein to register the changes.

Figure 1.27: Applying the changes

Alternatively, by adding a new entry in the Windows registry, you can quickly enable the **ADMIN$** share. The steps for the same are discussed hereunder:

1. In Run prompt type **regedit** to open registry editor.
2. Browse through the following sub key:

   HKEY_LOCAL_MACHINE\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\POLICIES\SYSTEM

3. Create a new entry with the below information
    - Key Name : LocalAccountTokenFilterPolicy
    - Key Type : DWORD (32-bit)
    - Key Value : 1
4. Exit registry editor.

Note:

As with any change to the registry, ensure that the above-mentioned change is also performed with utmost care, so as to avoid problems in the functioning of the operating system.

1.6.2 Configuring Windows Firewalls to Allow File and Print Sharing

In the case of virtual machines operating on Windows XP/Windows 2003/Windows 2008/Windows Vista/Windows 7, the firewall on the guest should be explicitly configured to allow Windows File and Print Sharing services which are required for the eG agent on the ESX host to communicate with the guest operating system.

To achieve this, do the following:

1. Open the Virtual Infrastructure Client console, and from the tree-structure in its left pane, select the guest OS (Windows XP/Windows 2003/Windows Vista/Windows 2008/Windows 7) on which the firewall should be configured (see Figure 1.28).

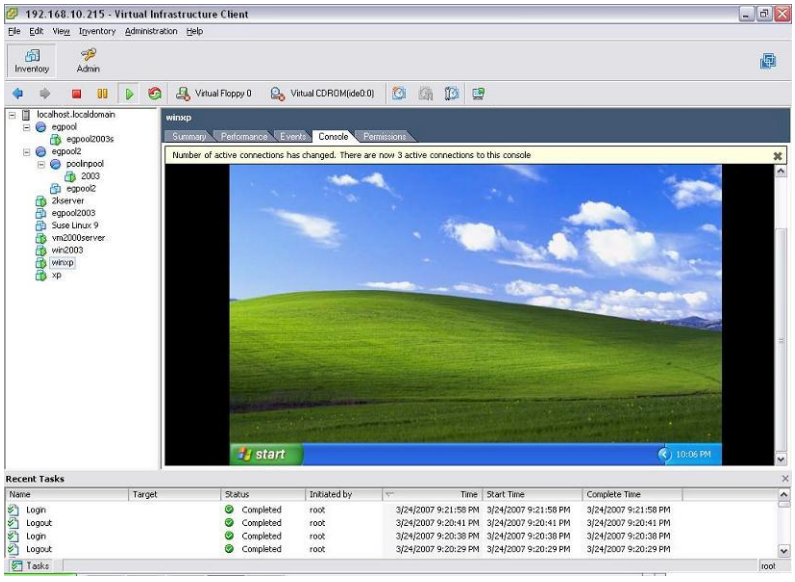


Figure 1.28: Selecting the guest OS

2. Follow the menu sequence: Start -> All Programs -> Control Panel (see Figure 1.29), and then double-click on the **Windows Firewall** option within.

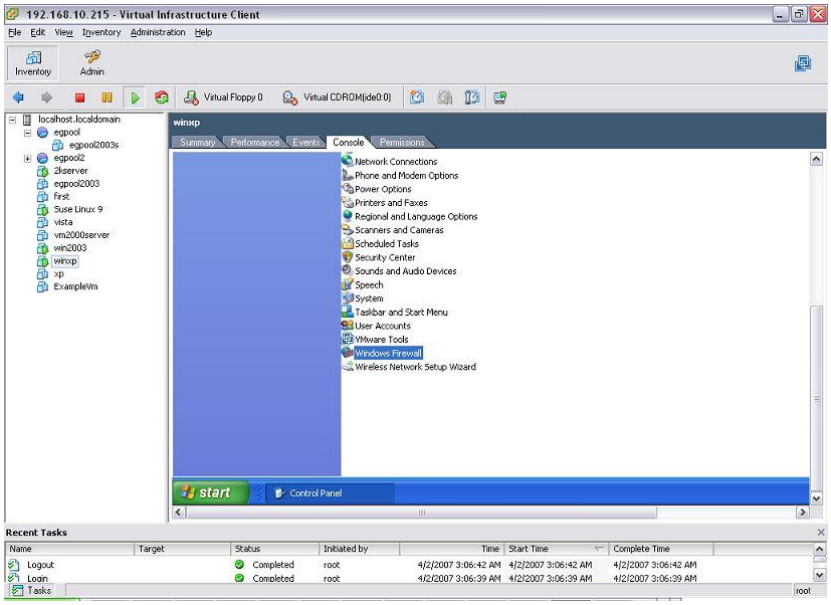


Figure 1.29: Opening the Windows Firewall

3. Figure 1.30 then appears, with the **General** tab selected by default.

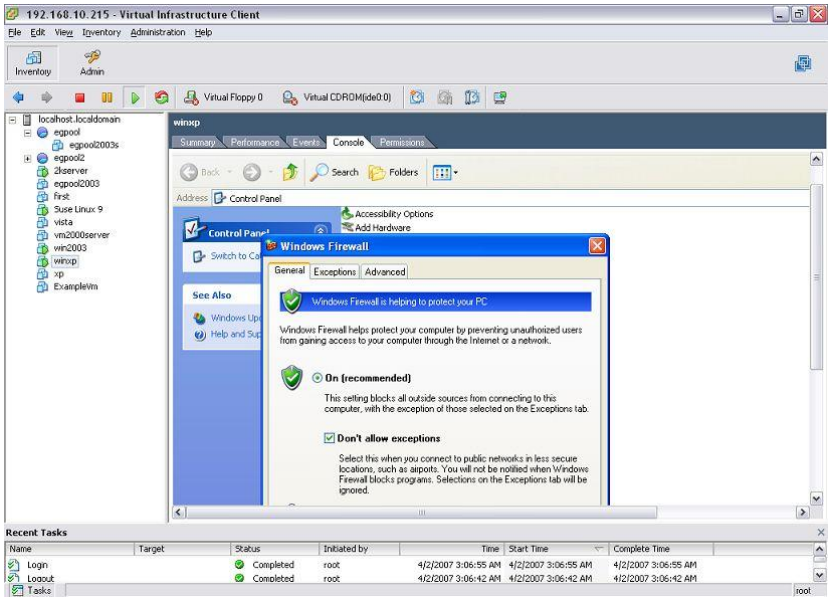


Figure 1.30: The General tab of the Windows Firewall dialog box

4. Deselect the **Don't allow exceptions** check box as indicated by Figure 1.31.

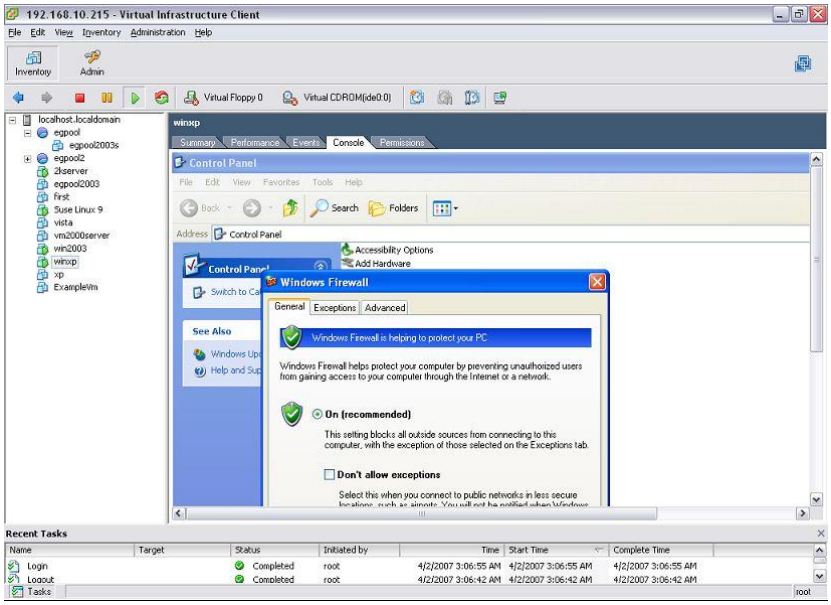


Figure 1.31: Deselecting the 'Don't allow exceptions' check box

5. Next, click on the **Exceptions** tab, and ensure that the **File and Printer Sharing** option is enabled (see Figure 1.32).

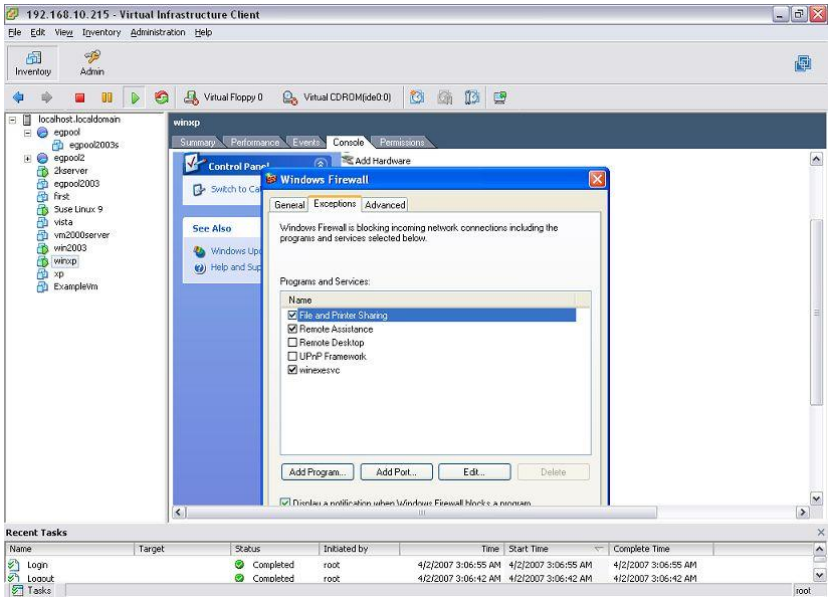


Figure 1.32: Enabling 'File and Printer Sharing'

6. Then, click the **Edit** button in Figure 1.33 to open the ports required for the agent-guest communication. Ensure that at least one of the listed TCP ports are enabled.

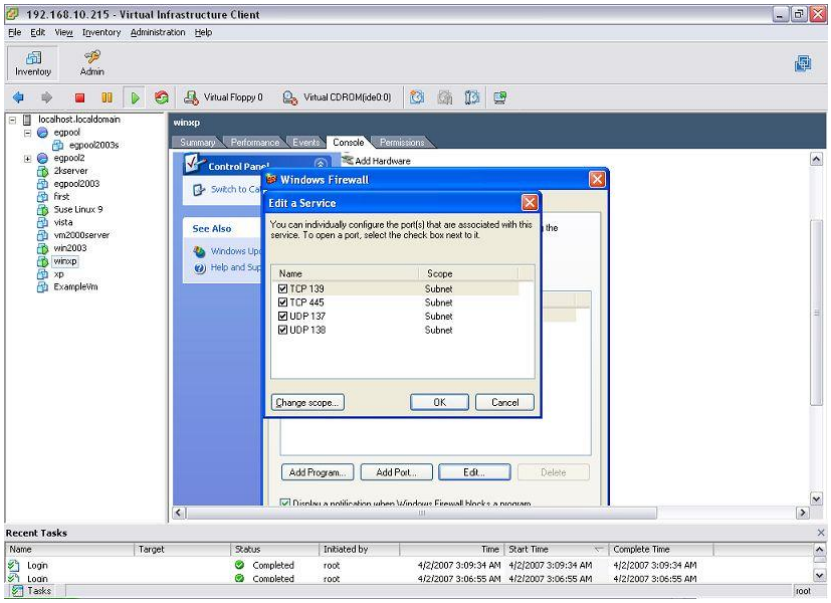


Figure 1.33: Opening ports

7. Finally, click the **OK** button to register the changes.

Chapter 2

Monitoring the Oracle VirtualBox

Figure 2.1 depicts the monitoring model of the Oracle VirtualBox.

Figure 2.1: The layer model of the Oracle VirtualBox

Each layer of Figure 2.1 is mapped to a wide variety of tests that report a wealth of performance tests using which administrators can find quick and accurate answers to the following performance queries:

- How many desktops are powered on simultaneously on the VirtualBox?
- 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 VirtualBox is a virtual guest running on?
- When was a guest moved from a VirtualBox? Which VirtualBox was the guest moved to?
- Why was the guest migrated? What activities on the VirtualBox caused the migration?

The section below will take a closer look at each layer of Figure 2.1.

2.1 The Operating System Layer

The tests mapped to this layer reveal the health of the VirtualBox host and hypervisor.

Figure 2.2: The tests mapped to the Operating System layer

The sections that follow will discuss the **Hypervisor Memory Details** test alone, as all other tests have been discussed in the *Monitoring Unix and Windows* servers document.

2.1.1 Hypervisor Memory Details Test

If one/more desktops on a VirtualBox are responding slowly to user requests, and the "inside view" of the desktops does not reveal anything untoward, then, you may want to focus on the measures reported by this test as they can indicate whether/not there is any contention for memory resources at the host-level. This is because, if the VirtualBox host has insufficient memory, it will adversely impact the memory allocations to the desktops on that VirtualBox, and consequently stall desktop operations.

This test closely monitors how the VirtualBox host and hypervisor use the physical memory resources, and warns administrators of potential memory crunches (if any).

Purpose	Closely monitors how the VirtualBox host and hypervisor use the physical memory resources, and warns administrators of potential memory crunches (if any)

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 –** Refers to the port used by the specified **HOST**. 4. **ORACLE HYPERVISOR USER** - Specify the name of the user who has the right to access the VirtualBox via SSH. 5. **ORACLE HYPERVISOR PASSWORD** - Provide the password of the **ORACLE HYPERVISOR USER**. 6. **CONFIRM PASSWORD** - Confirm the password by retyping it here. 7. **SUDOCMD** - This test executes certain privileged VDA (Virtual Desktop Access) commands to pull out the desired metrics from the VirtualBox. To enable the test to run these commands, you first need to install a **sudo** package on the VirtualBox host. The procedure for installing this package is detailed in Section 1.4.1.2 of this document. Once the package is installed, you need to specify the full path to the install directory of the **sudo** package in the **SUDOCMD** text box.		
Outputs of the test	One set of results for the Oracle VirtualBox being monitored		
Measurements made by the test	**Measurement**	**Measurement Unit**	**Interpretation**
	Total physical memory: Indicates the total amount of physical memory installed on Oracle VirtualBox.	MB	
	Used physical memory: Indicates the amount of physical memory currently used by the VirtualBox.	MB	Ideally, the value of this measure should be low.
	Free physical memory: Indicates the amount of unused physical memory on the VirtualBox.	MB	Ideally, the value of this measure should be high. A consistent decrease in the value of this measure is indicative of a steady memory erosion, which if left unattended, can significantly impact desktop functioning.
	Physical memory utilization: Indicates the percentage of physical memory utilized by the VirtualBox.	Percent	A low value is desired for this measure. If the value of this measure grows very close to 100%, it indicates that the VirtualBox will soon run out of physical memory resources. Such an occurrence can prove to be detrimental to not only the health of the VirtualBox, but also the desktops operating on it.

	Used hypervisor Physical memory: Indicates the total amount of physical memory currently being used by the hypervisor.	MB	If the **Physical memory utilization** measure reports a very high value, take a look at the value reported by this measure to determine whether the processes executing on the hypervisor are consuming too much physical memory.
	Free hypervisor physical memory: Indicates the total amount of physical memory free inside the hypervisor.	MB	A high value is desired for this measure. If the measure reports a very low value, it indicates that the hypervisor does not have adequate memory for its operations.

	Ballooned hypervisor physical memory: Indicates the total physical memory ballooned by the hypervisor.	MB	Normally, to change the amount of memory allocated to a virtual machine, one has to shut down the virtual machine entirely and modify its settings. With memory ballooning, memory that was allocated for a virtual machine can be given to another virtual machine without having to shut the machine down. When memory ballooning is requested, the VirtualBox allocates physical memory from the guest operating system on the kernel level and locks this memory down in the guest. This ensures that the guest will not use that memory any longer; no guest applications can allocate it, and the guest kernel will not use it either. VirtualBox can then re-use this memory and give it to another virtual machine. The memory made available through the ballooning mechanism is only available for re-use by VirtualBox. It is *not* returned as free memory to the host. Requesting balloon memory from a running guest will therefore not increase the amount of free, unallocated memory on the host. Effectively, memory ballooning is therefore a memory overcommitment mechanism for multiple virtual machines while they are running. This can be useful to temporarily start another machine, or in more complicated environments, for sophisticated memory management of many virtual machines that may be running in parallel depending on how memory is used by the guests.

	Shared memory: Indicates the total physical memory shared between the VMs.	MB	Shared Memory is a feature that enables more desktops to run on Oracle VDI Hypervisor hosts. By specifying an amount of memory to be shared between desktops, the Oracle VDI hypervisor host's memory can be automatically redistributed between desktops as required. The memory sharing percentage is the amount of memory that can be used for other desktops if a desktop does not require the full amount of memory for itself. For instance, if the desktop memory size is 1 GB and memory sharing is set to 40%, the desktop will initially have around 600 MB of real memory. The other 400 MB will be made available to the desktop on demand.

2.1.2 Processor Details Test

This test monitors the usage of processors by the Oracle VirtualBox, and thus reveals whether/not the host is consuming CPU resources optimally. With the help of this test, you can quickly detect any abnormal increase in CPU consumption by the VirtualBox host, determine the root-cause for the surge (is it owing to resource-intensive kernel processes? user processes?), and take rapid remedial measures, so that the problem can be contained before it affects the performance of the desktops operating on that VirtualBox.

Purpose	Monitors the usage of processors by the Oracle VirtualBox, and thus reveals whether/not the host is consuming CPU resources optimally
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 –** Refers to the port used by the specified **HOST**. 4. **ORACLE HYPERVISOR USER** - Specify the name of the user who has the right to access the VirtualBox via SSH. 5. **ORACLE HYPERVISOR PASSWORD** - Provide the password of the **ORACLE HYPERVISOR USER**. 6. **CONFIRM PASSWORD** - Confirm the password by retyping it here. 7. **SUDOCMD** - This test executes certain privileged VDA (Virtual Desktop Access) commands to pull out the desired metrics from the VirtualBox. To enable the test to run these commands, you first need to install a **sudo** package on the VirtualBox host. The procedure for installing this package is detailed in Section 1.4.1.2 of this document. Once the package is installed, you need to specify the full path to the install directory of the **sudo** package in the **SUDOCMD** text box.

Outputs of the test	One set of results for the Oracle VirtualBox being monitored		
Measurements made by the test	**Measurement**	**Measurement Unit**	**Interpretation**
	Total number of processors: Indicates the total number of processors assigned to the Oracle VirtualBox.	Number	
	Online processors: Indicates the number of processors currently utilized by the VirtualBox.	Number	
	CPU utilization: Indicates the percentage of processor time spent on kernel-level and user processes.	Percent	Ideally, the value of this measure should be low. A high value is indicative of excessive CPU consumption by the VirtualBox. Under such circumstances, you can use the values reported by the *Average user CPU utilization* and the *Average system CPU utilization* measures to determine where the processor time was spent the most - on executing kernel-level processes or user-level processes?
	Average user CPU utilization: Indicates the percentage of processor time spent on user processes.	Percent	If the CPU usage of the VirtualBox is very high, then the value of this measure will enable you to figure out whether CPU-intensive user processes are contributing to the CPU drain.
	Average system CPU utilization: Indicates the percentage of processor time spent on kernel-level processing.	Percent	If the CPU usage of the VirtualBox is very high, then the value of this measure will enable you to figure out whether CPU-intensive system processes are contributing to the CPU drain.
	Average idle CPU: Indicates the percentage of processor time spent idling.	Percent	A high value for this measure indicates that the processors have been idle for too long a time.

2.2 The Network Layer

Know whether the VirtualBox is available over the network or not, and promptly detect network traffic congestions using the tests mapped to this layer. Since both the tests depicted by Figure 2.3 have been dealt with in the *Monitoring Unix and Windows Servers* document, let us proceed to the next layer.

Figure 2.3: The tests mapped to the Network layer

2.3 The Application Processes Layer

VBoxSVC is the VirtualBox service process. It keeps track of all virtual machines that are running on the host. It is started automatically when the first guest boots. The **Processes** test mapped to this layer monitors the availability and resource usage of this process and reports abnormalities (if any).

Figure 2.4: The Application Processes layer

Since the *Monitoring Unix and Windows Servers* document already explains how to configure the **Processes** test and the measures it reports, let us proceed to the next layer.

2.4 The Oracle VDI VMs Layer

The **Oracle VDI 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, etc.) 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 VirtualBox, so as to minimize the impact it has on the other guests on the current VirtualBox.
- 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

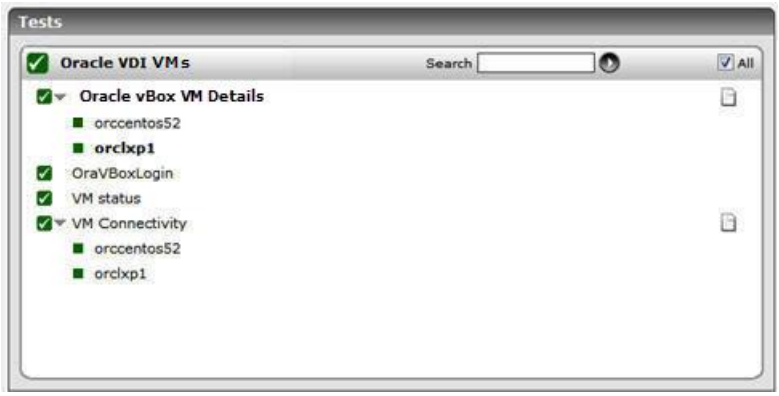


Figure 2.5: The tests mapped to the Oracle VDI VMs layer

2.4.1 Oracle vBox VM Details Test

This test auto-discovers the desktops operating on a VirtualBox and reports the powered-on state, the resource allocations, and the resource usage of each discovered desktop. In the process, the test promptly points administrators to unavailable or resource-intensive desktops.

Purpose	Auto-discovers the desktops operating on a VirtualBox and reports the powered-on state, the resource allocations, and the resource usage of each discovered desktop. In the process, the test promptly points administrators to unavailable or resource-intensive desktops

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 –** Refers to the port used by the specified **HOST**. 4. **ORACLE HYPERVISOR USER** - Specify the name of the user who has the right to access the VirtualBox via SSH. 5. **ORACLE HYPERVISOR PASSWORD** - Provide the password of the **ORACLE HYPERVISOR USER**. 6. **CONFIRM PASSWORD** - Confirm the password by retyping it here. 7. **SUDOCMD** - This test executes certain privileged VDA (Virtual Desktop Access) commands to pull out the desired metrics from the VirtualBox. To enable the test to run these commands, you first need to install a **sudo** package on the VirtualBox host. The procedure for installing this package is detailed in Section 1.4.1.2 of this document. Once the package is installed, you need to specify the full path to the install directory of the **sudo** package in the **SUDOCMD** text box. 8. **IGNORE VMS INSIDE VIEW** - Administrators of some high security virtualized 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 an Oracle VirtualBox 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. **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 with Windows VMs in particular, 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 (Windows)** 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 1.5 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**, `, and **ADMIN PASSWORD** parameters to *none*.

12. **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. In order to obtain a remote connection, the test must be configured with user privileges that allow remote communication with the virtual guests. 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 guests do not belong to any domain (as in the case of Linux/Solaris guests)** : In this case, specify "none" in the **DOMAIN** field, and specify a local administrator account name in the **ADMIN USER** below.

      Prior to this, you need to ensure that the same local administrator account is available or is explicitly created on each of the virtual machines to be monitored. Then, proceed to provide the password of the **ADMIN USER** against **ADMIN PASSWORD**, and confirm the password by retyping it in the **CONFIRM PASSWORD** text box.

If key-based authentication is implemented between the eG agent and the SSH daemon of a Linux guest, then, in the **ADMIN USER** text box, enter the name of the user whose **<USER_HOME_DIR>** (on that Linux guest) contains a **.ssh** directory with the *public key file* named **authorized_keys**. The **ADMIN PASSWORD** in this case will be the *passphrase* of the *public key*; the default *public key file* that is bundled with the eG agent takes the password *eginnovations*. Specify this as the **ADMIN PASSWORD** if you are using the default private/public key pair that is bundled with the eG agent to implement key-based authentication. On the other hand, if you are generating a new public/private key pair for this purpose, then use the *passphrase* that you provide while generating the pair. For the detailed procedure on *Implementing Key-based Authentication* refer to Section 2.6 of this document.

- **If the guests 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 2.4.1.1.1 of this document.
- **If the INSIDE VIEW USING flag is set to 'eG VM Agent (Windows)'** - In this case, the inside view can be obtained without domain administrator privileges. Therefore, set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.

13. **REPORT BY USER** - While monitoring a VirtualBox, the **REPORT BY USER** flag is set to **Yes** by default, indicating that by default, the guest operating systems on the VirtualBox are identified using the login of the user who is accessing the guest OS. In other words, this test will, by default, report measures for every *username_on_virtualmachinename*. If this flag is set to **No**, then the guests will be identified using the host name of the guest OS. In this case, the test will report measures for every *virtualmachinename*.

<table>
<tr><td></td><td colspan="3">14. 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.

15. 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.</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for each desktop on the Oracle VirtualBox being monitored</td></tr>
<tr><td>Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
<tr><td></td><td>Is virtual machine powered on?:

Indicates whether this desktop is powered-on or not currently.</td><td></td><td>The table below displays the States that can be reported by this measure, and their numeric equivalents:
<table><tr><th>State</th><th>Value</th></tr><tr><td>Off</td><td>0</td></tr><tr><td>On</td><td>1</td></tr></table>
Note:

By default, this measure reports one of the States listed in the table above. The graph of this measure however will represent the VM status using the numeric equivalents - '0' or '1'.</td></tr>
<tr><td></td><td>Total RAM size:

Indicates the amount of memory allocated to this desktop.</td><td>MB</td><td></td></tr>
</table>

	RAM utilization: Indicates the percentage of allocated RAM been utilized by this desktop.	Percent	Ideally, the value of this measure should be low. A high value is indicative of excessive memory consumption by a desktop. In the event of a slowdown, you can compare the value of this measure across desktops to identify the desktop that is memory-intensive, and could hence be causing the slowdown.
	Video memory: Indicates the total amount of video memory allocated to this desktop.	MB	This sets the size of the memory provided by the virtual graphics card available to the guest, in MB. As with the main memory, the specified amount will be allocated from the host's resident memory. Based on the amount of video memory, higher resolutions and color depths may be available.

	Balloon memory: Indicates the configured balloon memory for this desktop.	MB	Normally, to change the amount of memory allocated to a virtual machine, one has to shut down the virtual machine entirely and modify its settings. With memory ballooning, memory that was allocated for a virtual machine can be given to another virtual machine without having to shut the machine down. When memory ballooning is requested, the VirtualBox allocates physical memory from the guest operating system on the kernel level and locks this memory down in the guest. This ensures that the guest will not use that memory any longer; no guest applications can allocate it, and the guest kernel will not use it either. VirtualBox can then re-use this memory and give it to another virtual machine. The memory made available through the ballooning mechanism is only available for re-use by VirtualBox. It is *not* returned as free memory to the host. Requesting balloon memory from a running guest will therefore not increase the amount of free, unallocated memory on the host. Effectively, memory ballooning is therefore a memory overcommitment mechanism for multiple virtual machines while they are running. This can be useful to temporarily start another machine, or in more complicated environments, for sophisticated memory management of many virtual machines that may be running in parallel depending on how memory is used by the guests.
	Number of VCPUs: Indicates the number of virtual CPUs allocated to this desktop.	Number	
	User CPU utilization: Indicates the percentage of CPU time spent by this desktop on user processes.	Percent	If the *CPU utilization* of a desktop appears to be increasing consistently, then, you can use the value of this measure to figure out where the CPU time is being spent - on system processes? or user processes?

	Kernel CPU utilization: Indicates the percentage of CPU time spent by this desktop on kernel processes.	Percent	If the *CPU utilization* of a desktop appears to be increasing consistently, then, you can use the value of this measure to figure out where the CPU time is being spent - on kernel processes? or user processes?
	CPU utilization: Indicates the percentage of the allocated CPU resources used by this desktop.	Percent	Compare the value of this measure to identify the desktop that is running CPU-intensive applications. Once the desktop is identified, then, you can use the *User CPU utilization* and the *Kernel CPU utilization* measures to know where the desktop is spending maximum CPU time - on kernel processes? or user processes?

2.4.1.1.1 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.

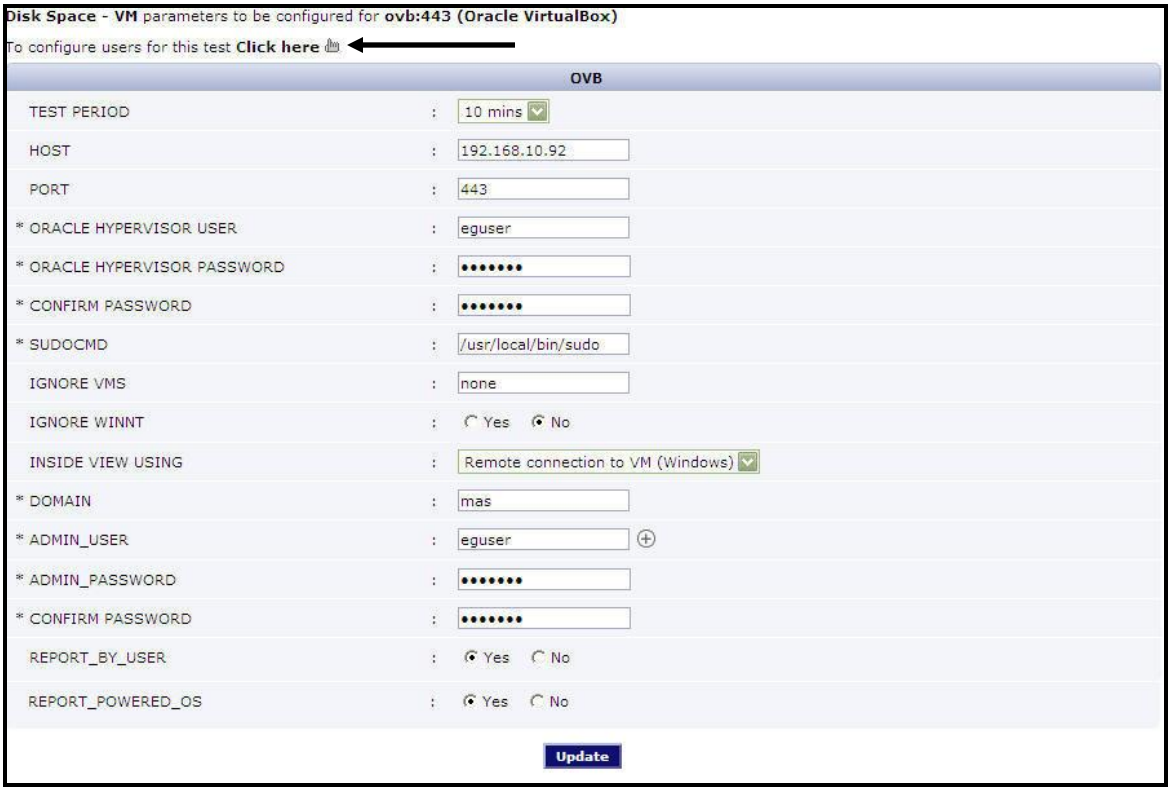


Figure 2.6: Configuring a VM test

Upon clicking, Figure 2.7 will appear, using which the VM user details can be configured.

Figure 2.7: The VM user configuration page

To add a user specification, do the following:

1. First, provide the name of the **Domain** to which the VMs belong (see Figure 2.7). If one/more VMs do not belong to any domain, then, specify *none* here.
2. 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. If key-based authentication is implemented between the eG agent and the SSH daemon of a Linux guest, then, in the **ADMIN USER** text box, enter the name of the user whose **<USER_HOME_DIR>** (on that Linux guest) contains a **.ssh** directory with the *public key file* named **authorized_keys**. The **ADMIN PASSWORD** in this case will be the *passphrase* of the *public key*; the default *public key file* that is bundled with the eG agent takes the password *eginnovations*. Specify this as the **ADMIN PASSWORD** if you are using the default private/public key pair that is bundled with the eG agent to implement key-based authentication. On the other hand, if you are generating a new public/private key pair for this purpose, then use the *passphrase* that you provide while generating the pair. For the detailed procedure on *Implementing Key-based Authentication* refer to Section 2.6 of this document.
3. The password of the specified **Admin User** should be mentioned in the **Admin Pwd** text box.
4. Confirm the password by retyping it in the **Confirm Pwd** text box.
5. To add more users, click on the ⊕ button in Figure 2.7. This will allow you to add one more user specification as depicted by Figure 2.8.

Figure 2.8: Adding another user

6. 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 2.9.

Figure 2.9: 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 2.9, 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., *jadmin* in our example (see Figure 2.9). If the first login attempt itself is successful, then the agent will ignore the second **Admin User** name.

7. To clear all the user specifications, simply click the **Clear** button in Figure 2.9.
8. To remove the details of a particular user alone, just click the ⊖ button in Figure 2.9.
9. To save the specification, just click on the **Update** button in Figure 2.9. 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.

2.4.2 Oracle VDI Logins Test

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

Purpose	Monitors the user logins to guests and reports the total count of logins and logouts

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 –** Refers to the port used by the specified **HOST**. 4. **ORACLE HYPERVISOR USER** - Specify the name of the user who has the right to access the VirtualBox via SSH. 5. **ORACLE HYPERVISOR PASSWORD** - Provide the password of the **ORACLE HYPERVISOR USER**. 6. **CONFIRM PASSWORD** - Confirm the password by retyping it here. 7. **SUDOCMD** - This test executes certain privileged VDA (Virtual Desktop Access) commands to pull out the desired metrics from the VirtualBox. To enable the test to run these commands, you first need to install a **sudo** package on the VirtualBox host. The procedure for installing this package is detailed in Section 1.4.1.2 of this document. Once the package is installed, you need to specify the full path to the install directory of the **sudo** package in the **SUDOCMD** text box. 8. **IGNORE VMS INSIDE VIEW** - Administrators of some high security virtualized 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 an Oracle VirtualBox 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. **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 with Windows VMs in particular, 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 (Windows)** 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 1.5 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*.

12. **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. In order to obtain a remote connection, the test must be configured with user privileges that allow remote communication with the virtual guests. 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 guests do not belong to any domain (as in the case of Linux/Solaris guests)** : In this case, specify "none" in the **DOMAIN** field, and specify a local administrator account name in the **ADMIN USER** below.

      Prior to this, you need to ensure that the same local administrator account is available or is explicitly created on each of the virtual machines to be monitored. Then, proceed to provide the password of the **ADMIN USER** against **ADMIN PASSWORD**, and confirm the password by retyping it in the **CONFIRM PASSWORD** text box.

If key-based authentication is implemented between the eG agent and the SSH daemon of a Linux guest, then, in the **ADMIN USER** text box, enter the name of the user whose **<USER_HOME_DIR>** (on that Linux guest) contains a **.ssh** directory with the *public key file* named **authorized_keys**. The **ADMIN PASSWORD** in this case will be the *passphrase* of the *public key*; the default *public key file* that is bundled with the eG agent takes the password *eginnovations*. Specify this as the **ADMIN PASSWORD** if you are using the default private/public key pair that is bundled with the eG agent to implement key-based authentication. On the other hand, if you are generating a new public/private key pair for this purpose, then use the *passphrase* that you provide while generating the pair. For the detailed procedure on *Implementing Key-based Authentication* refer to Section 2.6 of this document.

- **If the guests 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 2.4.1.1.1 of this document.

- **If the INSIDE VIEW USING flag is set to 'eG VM Agent (Windows)'** - In this case, the inside view can be obtained without domain administrator privileges. Therefore, set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.

13. **REPORT BY USER** - While monitoring a VirtualBox, the **REPORT BY USER** flag is set to **Yes** by default, indicating that by default, the guest operating systems on the VirtualBox are identified using the login of the user who is accessing the guest OS. In other words, this test will, by default, report measures for every *username_on_virtualmachinename*. If this flag is set to **No**, then the guests will be identified using the host name of the guest OS. In this case, the test will report measures for every *virtualmachinename*.

14. **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.

<table>
<tr><td></td><td colspan="3">15. 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.
16. 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.</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for the Oracle VirtualBox being monitored</td></tr>
<tr><td rowspan="4">Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
<tr><td>Current sessions:
Indicates the number of user sessions that are currently active across all guests</td><td>Number</td><td>This is a good indicator of the session load on the guests.</td></tr>
<tr><td>New logins:
Indicates the number of new logins to the guests.</td><td>Number</td><td>A consistent zero value could indicate a connection issue.</td></tr>
<tr><td>Percent new logins:
Indicates the percentage of current sessions that logged in during the last measurement period.</td><td>Percent</td><td></td></tr>
<tr><td></td><td>Sessions logging out:
Indicates the number of sessions that logged out.</td><td>Number</td><td>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.</td></tr>
</table>

2.4.3 VM Status Test

Whenever users complaint of inaccessibility of their desktops, administrators need to promptly determine the reason for the same - is it because the desktops are not running currently? is it because they are not even registered? or is it because they have been moved to another VirtualBox? The **VM**

Status test provides administrators with this information. This test tracks the status and movement of each desktop on the target VirtualBox, reports the number and names of desktops in various states, and also captures the migration of desktops to other VirtualBoxes.

Purpose	Tracks the status and movement of each desktop on the target VirtualBox, reports the number and names of desktops in various states, and also captures the migration of desktops to other VirtualBoxes
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 –** Refers to the port used by the specified **HOST**. 4. **ORACLE HYPERVISOR USER** - Specify the name of the user who has the right to access the VirtualBox via SSH. 5. **ORACLE HYPERVISOR PASSWORD** - Provide the password of the **ORACLE HYPERVISOR USER**. 6. **CONFIRM PASSWORD** - Confirm the password by retyping it here. 7. **SUDOCMD** - This test executes certain privileged VDA (Virtual Desktop Access) commands to pull out the desired metrics from the VirtualBox. To enable the test to run these commands, you first need to install a **sudo** package on the VirtualBox host. The procedure for installing this package is detailed in Section 1.4.1.2 of this document. Once the package is installed, you need to specify the full path to the install directory of the **sudo** package in the **SUDOCMD** text box. 8. **REPORT BY USER** - While monitoring a VirtualBox, the **REPORT BY USER** flag is set to **Yes** by default, indicating that by default, the guest operating systems on the VirtualBox are identified using the login of the user who is accessing the guest OS. In other words, this test will, by default, report measures for every *username_on_virtualmachinename*. If this flag is set to **No**, then the guests will be identified using the host name of the guest OS. In this case, the test will report measures for every *virtualmachinename*. 9. **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. 10. **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.**

	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 abnormal frequencies configured for the detailed diagnosis measures should not be 0.		
Outputs of the test	One set of results for the Oracle VirtualBox being monitored		
Measurements made by the test	**Measurement**	**Measurement Unit**	**Interpretation**
	Registered guests: Indicates the number of registered desktops on the Oracle VirtualBox.	Number	
	Running guests: Indicates the number of desktops currently running on the Oracle VirtualBox.	Number	
	Not running guests: Indicates the number of desktops currently not running on the Oracle VirtualBox.	Number	
	Added guests: Indicates the number of desktops newly added on the Oracle VirtualBox.	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 VirtualBox.
	Removed guests: Indicates the number of desktops newly removed from the Oracle VirtualBox.	Number	

2.4.4 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.

Purpose	Runs a connectivity check on each VM and reports whether the VM is accessible over the network or not

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 –** Refers to the port used by the specified **HOST**. 4. **ORACLE HYPERVISOR USER** - Specify the name of the user who has the right to access the VirtualBox via SSH. 5. **ORACLE HYPERVISOR PASSWORD** - Provide the password of the **ORACLE HYPERVISOR USER**. 6. **CONFIRM PASSWORD** - Confirm the password by retyping it here. 7. **SUDOCMD** - This test executes certain privileged VDA (Virtual Desktop Access) commands to pull out the desired metrics from the VirtualBox. To enable the test to run these commands, you first need to install a **sudo** package on the VirtualBox host. The procedure for installing this package is detailed in Section 1.4.1.2 of this document. Once the package is installed, you need to specify the full path to the install directory of the **sudo** package in the **SUDOCMD** text box. 8. **IGNORE VMS INSIDE VIEW** - Administrators of some high security virtualized 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 an Oracle VirtualBox 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. **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 with Windows VMs in particular, 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 (Windows)** 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 1.5 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*.

12. **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. In order to obtain a remote connection, the test must be configured with user privileges that allow remote communication with the virtual guests. 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 guests do not belong to any domain (as in the case of Linux/Solaris guests)** : In this case, specify "none" in the **DOMAIN** field, and specify a local administrator account name in the **ADMIN USER** below.

      Prior to this, you need to ensure that the same local administrator account is available or is explicitly created on each of the virtual machines to be monitored. Then, proceed to provide the password of the **ADMIN USER** against **ADMIN PASSWORD**, and confirm the password by retyping it in the **CONFIRM PASSWORD** text box.

<table>
<tr><td></td><td colspan="3">If key-based authentication is implemented between the eG agent and the SSH daemon of a Linux guest, then, in the ADMIN USER text box, enter the name of the user whose <USER_HOME_DIR> (on that Linux guest) contains a .ssh directory with the public key file named authorized_keys. The ADMIN PASSWORD in this case will be the passphrase of the public key; the default public key file that is bundled with the eG agent takes the password eginnovations. Specify this as the ADMIN PASSWORD if you are using the default private/public key pair that is bundled with the eG agent to implement key-based authentication. On the other hand, if you are generating a new public/private key pair for this purpose, then use the passphrase that you provide while generating the pair. For the detailed procedure on Implementing Key-based Authentication refer to Section 2.6 of this document.
➢ If the guests 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 2.4.1.1.1 of this document.
➢ If the INSIDE VIEW USING flag is set to 'eG VM Agent (Windows)' - In this case, the inside view can be obtained without domain administrator privileges. Therefore, set the DOMAIN, ADMIN USER, and ADMIN PASSWORD parameters to none.
13. PACKETSIZE - The size of packets used for the test (in bytes)
14. PACKETCOUNT - The number of packets to be transmitted during the test
15. TIMEOUT - How long after transmission should a packet be deemed lost (in seconds)
16. PACKETINTERVAL - Represents the interval (in milliseconds) between successive packet transmissions during the execution of the network test for a specific target.
17. 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.</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for each desktop on the Oracle VirtualBox being monitored</td></tr>
<tr><td>Measurements made by the</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
</table>

test	**Average delay:** Indicates the average delay between transmission of packet to a VM and receipt of the response to the packet at the source.	Secs	An increase in network latency could result from misconfiguration of the router(s) along the path, network congestion, retransmissions at the network, etc.
	Minimum 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: 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.

2.5 The Virtual Desktop Layer

The **Oracle VDI VMs** layer provides an "outside" view of the different desktops - the metrics reported at this layer are based on what the VirtualBox is seeing about the performance of the individual desktops. However, an outside view of the desktop and its applications may not be sufficient. For instance, suppose one of the disk partitions of the desktop 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 **Virtual Desktop** layer provide an "inside" view of the workings of each of the desktops - these tests execute on a VirtualBox host, but send probes into each of the virtual desktops to analyze how well each desktop 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 **Virtual Desktop** layer, does not display the list of tests associated with that layer. Instead, Figure 2.10 appears. This figure provides you with a list of all desktops and their respective state (see Figure 2.10).

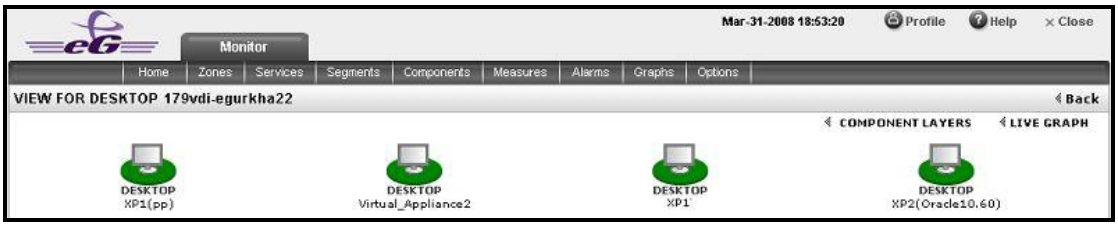


Figure 2.10: A list of desktops on an Oracle VirtualBox and their current state

To return to the layer model of the *Oracle VirtualBox* and view the tests associated with the **Virtual Desktop** layer, click on the **COMPONENT LAYERS** link in Figure 2.10. You can now view the list of tests mapped to the **Virtual Desktop** layer, as depicted by Figure 2.11 below.

Figure 2.11: The tests mapped to the Virtual Desktop layer

2.5.1 Disk Activity - VM Test

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

Purpose	To measure the input/output utilization of each physical disk on each desktop of an Oracle VirtualBox
Target of the test	An Oracle VirtualBox
Agent deploying the test	An internal/remote agent

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 –** Refers to the port used by the specified **HOST**. 4. **ORACLE HYPERVISOR USER** - Specify the name of the user who has the right to access the VirtualBox via SSH. 5. **ORACLE HYPERVISOR PASSWORD** - Provide the password of the **ORACLE HYPERVISOR USER**. 6. **CONFIRM PASSWORD** - Confirm the password by retyping it here. 7. **SUDOCMD** - This test executes certain privileged VDA (Virtual Desktop Access) commands to pull out the desired metrics from the VirtualBox. To enable the test to run these commands, you first need to install a **sudo** package on the VirtualBox host. The procedure for installing this package is detailed in Section 1.4.1.2 of this document. Once the package is installed, you need to specify the full path to the install directory of the **sudo** package in the **SUDOCMD** text box. 8. **IGNORE VMS INSIDE VIEW** - Administrators of some high security virtualized 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 an Oracle VirtualBox 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. **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 with Windows VMs in particular, 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 (Windows)** 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 1.5 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*.

12. **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. In order to obtain a remote connection, the test must be configured with user privileges that allow remote communication with the virtual guests. 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 guests do not belong to any domain (as in the case of Linux/Solaris guests)**: In this case, specify "none" in the **DOMAIN** field, and specify a local administrator account name in the **ADMIN USER** below.

      Prior to this, you need to ensure that the same local administrator account is available or is explicitly created on each of the virtual machines to be monitored. Then, proceed to provide the password of the **ADMIN USER** against **ADMIN PASSWORD**, and confirm the password by retyping it in the **CONFIRM PASSWORD** text box.

If key-based authentication is implemented between the eG agent and the SSH daemon of a Linux guest, then, in the **ADMIN USER** text box, enter the name of the user whose **<USER_HOME_DIR>** (on that Linux guest) contains a **.ssh** directory with the *public key file* named **authorized_keys**. The **ADMIN PASSWORD** in this case will be the *passphrase* of the *public key*; the default *public key file* that is bundled with the eG agent takes the password *eginnovations*. Specify this as the **ADMIN PASSWORD** if you are using the default private/public key pair that is bundled with the eG agent to implement key-based authentication. On the other hand, if you are generating a new public/private key pair for this purpose, then use the *passphrase* that you provide while generating the pair. For the detailed procedure on *Implementing Key-based Authentication* refer to Section 2.6 of this document.

- **If the guests 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 2.4.1.1.1 of this document.
- **If the INSIDE VIEW USING flag is set to 'eG VM Agent (Windows)'** - In this case, the inside view can be obtained without domain administrator privileges. Therefore, set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.

13. **REPORT BY USER** - While monitoring a VirtualBox, the **REPORT BY USER** flag is set to **Yes** by default, indicating that by default, the guest operating systems on the VirtualBox are identified using the login of the user who is accessing the guest OS. In other words, this test will, by default, report measures for every *username_on_virtualmachinename*. If this flag is set to **No**, then the guests will be identified using the host name of the guest OS. In this case, the test will report measures for every *virtualmachinename*.
14. **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.

	15. **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 combination of *virtual_guest:disk_partition* or *guest_user:disk_partition*.

Measurements made by the test	Measurement	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 percentage of elapsed time that the selected disk drive is busy servicing read requests.	Percent	
	Percent writes to virtual disk: Indicates the percentage 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.
	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 avarage 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. **The detailed diagnosis for this test is available for Windows guests only, and not Linux guests**.

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
Jan 29, 2008 10:35:10							
	232	services	208.48	102.90	2.34	105.57	2
Jan 29, 2008 10:24:55							
	232	services	208.10	102.72	2.33	105.38	2

Figure 2.12: The detailed diagnosis of the Percent virtual busy measure

2.5.2 Disk Space - VM Test

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

Purpose	To measure the space usage of every disk partition on each desktop of a VirtualBox
Target of the test	An Oracle VirtualBox

Agent deploying the test	An internal/remote agent
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 –** Refers to the port used by the specified **HOST**. 4. **ORACLE HYPERVISOR USER** - Specify the name of the user who has the right to access the VirtualBox via SSH. 5. **ORACLE HYPERVISOR PASSWORD** - Provide the password of the **ORACLE HYPERVISOR USER**. 6. **CONFIRM PASSWORD** - Confirm the password by retyping it here. 7. **SUDOCMD** - This test executes certain privileged VDA (Virtual Desktop Access) commands to pull out the desired metrics from the VirtualBox. To enable the test to run these commands, you first need to install a **sudo** package on the VirtualBox host. The procedure for installing this package is detailed in Section 1.4.1.2 of this document. Once the package is installed, you need to specify the full path to the install directory of the **sudo** package in the **SUDOCMD** text box. 8. **IGNORE VMS INSIDE VIEW** - Administrators of some high security virtualized 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 an Oracle VirtualBox 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. **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 with Windows VMs in particular, 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 (Windows)** 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 1.5 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*.

12. **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. In order to obtain a remote connection, the test must be configured with user privileges that allow remote communication with the virtual guests. 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 guests do not belong to any domain (as in the case of Linux/Solaris guests)** : In this case, specify "none" in the **DOMAIN** field, and specify a local administrator account name in the **ADMIN USER** below.

      Prior to this, you need to ensure that the same local administrator account is available or is explicitly created on each of the virtual machines to be monitored. Then, proceed to provide the password of the **ADMIN USER** against **ADMIN PASSWORD**, and confirm the password by retyping it in the **CONFIRM PASSWORD** text box.

	If key-based authentication is implemented between the eG agent and the SSH daemon of a Linux guest, then, in the **ADMIN USER** text box, enter the name of the user whose **<USER_HOME_DIR>** (on that Linux guest) contains a **.ssh** directory with the *public key file* named **authorized_keys**. The **ADMIN PASSWORD** in this case will be the *passphrase* of the *public key*; the default *public key file* that is bundled with the eG agent takes the password *eginnovations*. Specify this as the **ADMIN PASSWORD** if you are using the default private/public key pair that is bundled with the eG agent to implement key-based authentication. On the other hand, if you are generating a new public/private key pair for this purpose, then use the *passphrase* that you provide while generating the pair. For the detailed procedure on *Implementing Key-based Authentication* refer to Section 2.6 of this document. ➢ **If the guests 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 2.4.1.1.1 of this document. ➢ **If the INSIDE VIEW USING flag is set to 'eG VM Agent (Windows)'** - In this case, the inside view can be obtained without domain administrator privileges. Therefore, set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*. 13. **REPORT BY USER** - While monitoring a VirtualBox, the **REPORT BY USER** flag is set to **Yes** by default, indicating that by default, the guest operating systems on the VirtualBox are identified using the login of the user who is accessing the guest OS. In other words, this test will, by default, report measures for every *username_on_virtualmachinename*. If this flag is set to **No**, then the guests will be identified using the host name of the guest OS. In this case, the test will report measures for every *virtualmachinename*. 14. **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.
Outputs of the test	One set of results for every combination of *virtual_guest:disk_partition* or *guest_user:disk_partition*.

Measurements made by the test	Measurement	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.

2.5.3 System Details - VM Test

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

Purpose	To measure the CPU and memory usage of each desktop of an Oracle VirtualBox
Target of the	An Oracle VirtualBox

test	
Agent deploying the test	An internal/remote agent
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 –** Refers to the port used by the specified **HOST**. 4. **ORACLE HYPERVISOR USER** - Specify the name of the user who has the right to access the VirtualBox via SSH. 5. **ORACLE HYPERVISOR PASSWORD** - Provide the password of the **ORACLE HYPERVISOR USER**. 6. **CONFIRM PASSWORD** - Confirm the password by retyping it here. 7. **SUDOCMD** - This test executes certain privileged VDA (Virtual Desktop Access) commands to pull out the desired metrics from the VirtualBox. To enable the test to run these commands, you first need to install a **sudo** package on the VirtualBox host. The procedure for installing this package is detailed in Section 1.4.1.2 of this document. Once the package is installed, you need to specify the full path to the install directory of the **sudo** package in the **SUDOCMD** text box. 8. **IGNORE VMS INSIDE VIEW** - Administrators of some high security virtualized 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 an Oracle VirtualBox 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. **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 with Windows VMs in particular, 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 (Windows)** 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 1.5 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*.

12. **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. In order to obtain a remote connection, the test must be configured with user privileges that allow remote communication with the virtual guests. 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 guests do not belong to any domain (as in the case of Linux/Solaris guests)** : In this case, specify "none" in the **DOMAIN** field, and specify a local administrator account name in the **ADMIN USER** below.

  Prior to this, you need to ensure that the same local administrator account is available or is explicitly created on each of the virtual machines to be monitored. Then, proceed to provide the password of the **ADMIN USER** against **ADMIN PASSWORD**, and confirm the password by retyping it in the **CONFIRM PASSWORD** text box.

  If key-based authentication is implemented between the eG agent and the SSH daemon of a Linux guest, then, in the **ADMIN USER** text box, enter the name of the user whose **<USER_HOME_DIR>** (on that Linux guest) contains a **.ssh** directory with the *public key file* named **authorized_keys**. The **ADMIN PASSWORD** in this case will be the *passphrase* of the *public key*; the default *public key file* that is bundled with the eG agent takes the password *eginnovations*. Specify this as the **ADMIN PASSWORD** if you are using the default private/public key pair that is bundled with the eG agent to implement key-based authentication. On the other hand, if you are generating a new public/private key pair for this purpose, then use the *passphrase* that you provide while generating the pair. For the detailed procedure on *Implementing Key-based Authentication* refer to Section 2.6 of this document.

- **If the guests 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 2.4.1.1.1 of this document.

- **If the INSIDE VIEW USING flag is set to 'eG VM Agent (Windows)'** - In this case, the inside view can be obtained without domain administrator privileges. Therefore, set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.

13. **REPORT BY USER** - While monitoring a VirtualBox, the **REPORT BY USER** flag is set to **Yes** by default, indicating that by default, the guest operating systems on the VirtualBox are identified using the login of the user who is accessing the guest OS. In other words, this test will, by default, report measures for every *username_on_virtualmachinename*. If this flag is set to **No**, then the guests will be identified using the host name of the guest OS. In this case, the test will report measures for every *virtualmachinename*.

<table>
<tr><td></td><td colspan="3">14. 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.

15. USE TOP FOR DD - This parameter is applicable only to Linux VMs. By default, this parameter is set to No. This indicates that, by default, this test will report the detailed diagnosis of the Virtual CPU utilization measure for each processor on a Linux VM by executing the usr/bin/ps command. On some Linux flavors however, this command may not function properly. In such cases, set the USE TOP FOR DD parameter to Yes. This will enable the eG agent to extract the detailed diagnosis of the Virtual CPU utilization measure by executing the /usr/bin/top command instead.

16. 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.

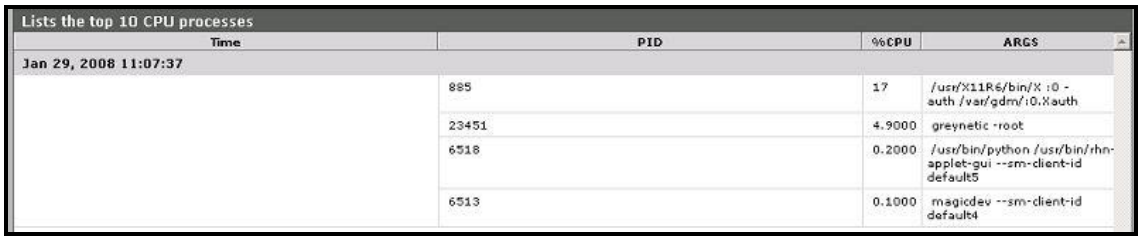
The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:
➢ The eG manager license should allow the detailed diagnosis capability
➢ Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for every combination of virtual_guest:processor or guest_user:processor.</td></tr>
<tr><td>Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
<tr><td></td><td>Virtual CPU utilization:
This measurement indicates the percentage of CPU utilized by the processor.</td><td>Percent</td><td>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.</td></tr>
<tr><td></td><td>System usage of virtual CPU:
Indicates the percentage of CPU time spent for system-level processing.</td><td>Percent</td><td>An unusually high value indicates a problem and may be due to too many system-level tasks executing simultaneously.</td></tr>
</table>

	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	A very low value of free memory is also an indication of high memory utilization on a guest. The detailed diagnosis of this measure, if enabled, lists the top 10 processes responsible for maximum memory consumption on the guest.
	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 Figure 2.13). 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

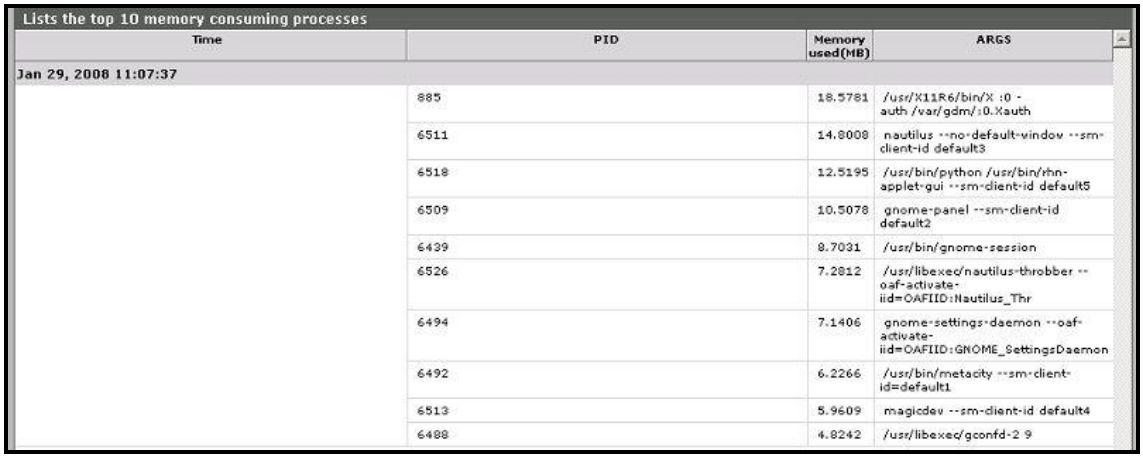
Time	PID	%CPU	ARGS
Jan 29, 2008 11:07:37			
	885	17	/usr/X11R6/bin/X :0 -auth /var/gdm/:0.Xauth
	23451	4.9000	greynetic -root
	6518	0.2000	/usr/bin/python /usr/bin/rhn-applet-gui --sm-client-id default5
	6513	0.1000	magicdev --sm-client-id default4

Figure 2.13: 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 2.14). 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
Jan 29, 2008 11:07:37			
	885	18.5781	/usr/X11R6/bin/X :0 -auth /var/gdm/:0.Xauth
	6511	14.8008	nautilus --no-default-window --sm-client-id default3
	6518	12.5195	/usr/bin/python /usr/bin/rhn-applet-gui --sm-client-id default5
	6509	10.5078	gnome-panel --sm-client-id default2
	6439	8.7031	/usr/bin/gnome-session
	6526	7.2812	/usr/libexec/nautilus-throbber --oaf-activate-iid=OAFIID:Nautilus_Thr
	6494	7.1406	gnome-settings-daemon --oaf-activate-iid=OAFIID:GNOME_SettingsDaemon
	6492	6.2266	/usr/bin/metacity --sm-client-id=default1
	6513	5.9609	magicdev --sm-client-id default4
	6488	4.8242	/usr/libexec/gconfd-2 9

Figure 2.14: The detailed diagnosis of the Free memory measure listing the top 10 memory consuming processes

2.5.4 Uptime - VM Test

In most virtualized environments, it is essential to monitor the uptime of desktops hosting critical server applications in the infrastructure. By tracking the uptime of each of the desktops, administrators can determine what percentage of time a desktop 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 desktop. By knowing that a specific desktop has been up for an unusually long time, an administrator may come to know that the scheduled reboot task is not working on a desktop.

The Uptime - VM test included in the eG agent monitors the uptime of each desktop on an Oracle VirtualBox.

Purpose	To monitor the uptime of each desktop on an Oracle VirtualBox
Target of the test	An Oracle VirtualBox
Agent deploying the test	An internal/remote agent
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 –** Refers to the port used by the specified **HOST**. 4. **ORACLE HYPERVISOR USER** - Specify the name of the user who has the right to access the VirtualBox via SSH. 5. **ORACLE HYPERVISOR PASSWORD** - Provide the password of the **ORACLE HYPERVISOR USER**. 6. **CONFIRM PASSWORD** - Confirm the password by retyping it here. 7. **SUDOCMD** - This test executes certain privileged VDA (Virtual Desktop Access) commands to pull out the desired metrics from the VirtualBox. To enable the test to run these commands, you first need to install a **sudo** package on the VirtualBox host. The procedure for installing this package is detailed in Section 1.4.1.2 of this document. Once the package is installed, you need to specify the full path to the install directory of the **sudo** package in the **SUDOCMD** text box. 8. **IGNORE VMS INSIDE VIEW** - Administrators of some high security virtualized 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 an Oracle VirtualBox 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. **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 with Windows VMs in particular, 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 (Windows)** 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 1.5 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*.

12. **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. In order to obtain a remote connection, the test must be configured with user privileges that allow remote communication with the virtual guests. 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 guests do not belong to any domain (as in the case of Linux/Solaris guests)** : In this case, specify "none" in the **DOMAIN** field, and specify a local administrator account name in the **ADMIN USER** below.

  Prior to this, you need to ensure that the same local administrator account is available or is explicitly created on each of the virtual machines to be monitored. Then, proceed to provide the password of the **ADMIN USER** against **ADMIN PASSWORD**, and confirm the password by retyping it in the **CONFIRM PASSWORD** text box.

  If key-based authentication is implemented between the eG agent and the SSH daemon of a Linux guest, then, in the **ADMIN USER** text box, enter the name of the user whose **<USER_HOME_DIR>** (on that Linux guest) contains a **.ssh** directory with the *public key file* named **authorized_keys**. The **ADMIN PASSWORD** in this case will be the *passphrase* of the *public key*; the default *public key file* that is bundled with the eG agent takes the password *eginnovations*. Specify this as the **ADMIN PASSWORD** if you are using the default private/public key pair that is bundled with the eG agent to implement key-based authentication. On the other hand, if you are generating a new public/private key pair for this purpose, then use the *passphrase* that you provide while generating the pair. For the detailed procedure on *Implementing Key-based Authentication* refer to Section 2.6 of this document.

- **If the guests 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 2.4.1.1.1 of this document.

- **If the INSIDE VIEW USING flag is set to 'eG VM Agent (Windows)'** - In this case, the inside view can be obtained without domain administrator privileges. Therefore, set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.

13. **REPORT BY USER** - While monitoring a VirtualBox, the **REPORT BY USER** flag is set to **Yes** by default, indicating that by default, the guest operating systems on the VirtualBox are identified using the login of the user who is accessing the guest OS. In other words, this test will, by default, report measures for every *username_on_virtualmachinename*. If this flag is set to **No**, then the guests will be identified using the host name of the guest OS. In this case, the test will report measures for every *virtualmachinename*.

	14. **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. 15. **REPORTMANAGERTIME** - By default, this flag is set to **Yes**, indicating that, by default, the detailed diagnosis of this test, if enabled, will report the shutdown and reboot times of the VMs in the manager's time zone. If this flag is set to **No**, then the shutdown and reboot times are shown in the time zone of the system where the agent is running (i.e., the system being managed for agent-based monitoring, and the system on which the remote agent is running - for agentless monitoring). 16. **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: 17. The eG manager license should allow the detailed diagnosis capability 18. Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.
Output of the Test	One set of results for each desktop discovered on the VirtualBox being monitored

Measurements made by the test	**Measurement**	**Measurement Unit**	**Interpretation**
	Rebooted: Indicates whether this guest 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: Indicates the time period that the guest 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 VM 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 VM 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: Indicates the total time that the guest has been up since its last reboot.	Mins	Administrators may wish to be alerted if a VM has been running without a reboot for a very long period. Setting a threshold for this metric allows administrators to determine such conditions.

Note:

If a value less than a minute is configured as the **TEST PERIOD** of the Uptime - VM test, then, the **Uptime during the last measure period** measure will report the value 0 for Unix VMs (only) until the minute boundary is crossed. For instance, if you configure the Uptime - VM test to run every 10 seconds, then, for the first 5 test execution cyles (i.e., 10 x 5 = 50 seconds), the **Uptime during the last measure period** measure will report the value 0 for Unix VMs; however, the sixth time the test executes (i.e, when test execution touches the 1 minute boundary), this measure will report the value 60 seconds for the same VMs. Thereafter, every sixth measurement period will report 60 seconds as the uptime of the Unix VMs. This is because, Unix-based operating systems report uptime only in minutes and not in seconds.

2.5.5 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 - VM test tracks critical metrics corresponding to the System PTEs and the pool areas of kernel memory of a Windows desktop.

Purpose	Tracks critical metrics corresponding to the System PTEs and the pool areas of kernel memory of each Windows desktop of an Oracle VirtualBox
Target of the test	An Oracle VirtualBox
Agent deploying the test	An internal/remote agent
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 –** Refers to the port used by the specified **HOST**. 4. **ORACLE HYPERVISOR USER** - Specify the name of the user who has the right to access the VirtualBox via SSH. 5. **ORACLE HYPERVISOR PASSWORD** - Provide the password of the **ORACLE HYPERVISOR USER**. 6. **CONFIRM PASSWORD** - Confirm the password by retyping it here. 7. **SUDOCMD** - This test executes certain privileged VDA (Virtual Desktop Access) commands to pull out the desired metrics from the VirtualBox. To enable the test to run these commands, you first need to install a **sudo** package on the VirtualBox host. The procedure for installing this package is detailed in Section 1.4.1.2 of this document. Once the package is installed, you need to specify the full path to the install directory of the **sudo** package in the **SUDOCMD** text box. 8. **IGNORE VMS INSIDE VIEW** - Administrators of some high security virtualized 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 an Oracle VirtualBox 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. **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 with Windows VMs in particular, 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 (Windows)** 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 1.5 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*.

12. **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. In order to obtain a remote connection, the test must be configured with user privileges that allow remote communication with the virtual guests. 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 guests do not belong to any domain (as in the case of Linux/Solaris guests)** : In this case, specify "none" in the **DOMAIN** field, and specify a local administrator account name in the **ADMIN USER** below.

  Prior to this, you need to ensure that the same local administrator account is available or is explicitly created on each of the virtual machines to be monitored. Then, proceed to provide the password of the **ADMIN USER** against **ADMIN PASSWORD**, and confirm the password by retyping it in the **CONFIRM PASSWORD** text box.

  If key-based authentication is implemented between the eG agent and the SSH daemon of a Linux guest, then, in the **ADMIN USER** text box, enter the name of the user whose **<USER_HOME_DIR>** (on that Linux guest) contains a **.ssh** directory with the *public key file* named **authorized_keys**. The **ADMIN PASSWORD** in this case will be the *passphrase* of the *public key*; the default *public key file* that is bundled with the eG agent takes the password *eginnovations*. Specify this as the **ADMIN PASSWORD** if you are using the default private/public key pair that is bundled with the eG agent to implement key-based authentication. On the other hand, if you are generating a new public/private key pair for this purpose, then use the *passphrase* that you provide while generating the pair. For the detailed procedure on *Implementing Key-based Authentication* refer to Section 2.6 of this document.

- **If the guests 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 2.4.1.1.1 of this document.

- **If the INSIDE VIEW USING flag is set to 'eG VM Agent (Windows)'** - In this case, the inside view can be obtained without domain administrator privileges. Therefore, set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.

13. **REPORT BY USER** - While monitoring a VirtualBox, the **REPORT BY USER** flag is set to **Yes** by default, indicating that by default, the guest operating systems on the VirtualBox are identified using the login of the user who is accessing the guest OS. In other words, this test will, by default, report measures for every *username_on_virtualmachinename*. If this flag is set to **No**, then the guests will be identified using the host name of the guest OS. In this case, the test will report measures for every *virtualmachinename*.

	14. **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.		
Outputs of the test	One set of results for every Windows desktop on the monitored Oracle VirtualBox		
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.

2.5.6 Windows Network Traffic - VM Test

This is an internal test that monitors the incoming and outgoing traffic through each Windows desktop of an Oracle VirtualBox.

Purpose	To measure the incoming and outgoing traffic through each Windows desktop of an Oracle VirtualBox
Target of the test	An Oracle VirtualBox
Agent deploying the test	An internal/remote agent

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 –** Refers to the port used by the specified **HOST**. 4. **ORACLE HYPERVISOR USER** - Specify the name of the user who has the right to access the VirtualBox via SSH. 5. **ORACLE HYPERVISOR PASSWORD** - Provide the password of the **ORACLE HYPERVISOR USER**. 6. **CONFIRM PASSWORD** - Confirm the password by retyping it here. 7. **SUDOCMD** - This test executes certain privileged VDA (Virtual Desktop Access) commands to pull out the desired metrics from the VirtualBox. To enable the test to run these commands, you first need to install a **sudo** package on the VirtualBox host. The procedure for installing this package is detailed in Section 1.4.1.2 of this document. Once the package is installed, you need to specify the full path to the install directory of the **sudo** package in the **SUDOCMD** text box. 8. **IGNORE VMS INSIDE VIEW** - Administrators of some high security virtualized 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 an Oracle VirtualBox 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. **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 with Windows VMs in particular, 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 (Windows)** 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 1.5 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*.

12. **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. In order to obtain a remote connection, the test must be configured with user privileges that allow remote communication with the virtual guests. 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 guests do not belong to any domain (as in the case of Linux/Solaris guests)** : In this case, specify "none" in the **DOMAIN** field, and specify a local administrator account name in the **ADMIN USER** below.

      Prior to this, you need to ensure that the same local administrator account is available or is explicitly created on each of the virtual machines to be monitored. Then, proceed to provide the password of the **ADMIN USER** against **ADMIN PASSWORD**, and confirm the password by retyping it in the **CONFIRM PASSWORD** text box.

	If key-based authentication is implemented between the eG agent and the SSH daemon of a Linux guest, then, in the **ADMIN USER** text box, enter the name of the user whose **<USER_HOME_DIR>** (on that Linux guest) contains a **.ssh** directory with the *public key file* named **authorized_keys**. The **ADMIN PASSWORD** in this case will be the *passphrase* of the *public key*; the default *public key file* that is bundled with the eG agent takes the password *eginnovations*. Specify this as the **ADMIN PASSWORD** if you are using the default private/public key pair that is bundled with the eG agent to implement key-based authentication. On the other hand, if you are generating a new public/private key pair for this purpose, then use the *passphrase* that you provide while generating the pair. For the detailed procedure on *Implementing Key-based Authentication* refer to Section 2.6 of this document. ➢ **If the guests 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 2.4.1.1.1 of this document. ➢ **If the INSIDE VIEW USING flag is set to 'eG VM Agent (Windows)'** - In this case, the inside view can be obtained without domain administrator privileges. Therefore, set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*. 13. **REPORT BY USER** - While monitoring a VirtualBox, the **REPORT BY USER** flag is set to **Yes** by default, indicating that by default, the guest operating systems on the VirtualBox are identified using the login of the user who is accessing the guest OS. In other words, this test will, by default, report measures for every *username_on_virtualmachinename*. If this flag is set to **No**, then the guests will be identified using the host name of the guest OS. In this case, the test will report measures for every *virtualmachinename*. 14. **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.
Outputs of the test	One set of results for every *Windows_virtual_guest:network_interface* combination or *Windows_VM_guest_user:network_interface* combination

Measurements made by the test	Measurement	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	
	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 - VM test is not reporting measures for a guest, make sure that you have enabled the SNMP service for the guest.

2.5.7 Network Traffic - VM Test

This is an internal test that monitors the incoming and outgoing traffic through each Linux desktop on an Oracle VirtualBox.

Purpose	To measure the incoming and outgoing traffic through each Linux desktop on an Oracle VirtualBox
Target of the test	An Oracle VirtualBox
Agent deploying the test	An internal/remote agent
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 –** Refers to the port used by the specified **HOST**. 4. **ORACLE HYPERVISOR USER** - Specify the name of the user who has the right to access the VirtualBox via SSH. 5. **ORACLE HYPERVISOR PASSWORD** - Provide the password of the **ORACLE HYPERVISOR USER**. 6. **CONFIRM PASSWORD** - Confirm the password by retyping it here. 7. **SUDOCMD** - This test executes certain privileged VDA (Virtual Desktop Access) commands to pull out the desired metrics from the VirtualBox. To enable the test to run these commands, you first need to install a **sudo** package on the VirtualBox host. The procedure for installing this package is detailed in Section 1.4.1.2 of this document. Once the package is installed, you need to specify the full path to the install directory of the **sudo** package in the **SUDOCMD** text box. 8. **IGNORE VMS INSIDE VIEW** - Administrators of some high security virtualized 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 an Oracle VirtualBox 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. **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 with Windows VMs in particular, 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 (Windows)** 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 1.5 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*.

12. **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. In order to obtain a remote connection, the test must be configured with user privileges that allow remote communication with the virtual guests. 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 guests do not belong to any domain (as in the case of Linux/Solaris guests)** : In this case, specify "none" in the **DOMAIN** field, and specify a local administrator account name in the **ADMIN USER** below.

  Prior to this, you need to ensure that the same local administrator account is available or is explicitly created on each of the virtual machines to be monitored. Then, proceed to provide the password of the **ADMIN USER** against **ADMIN PASSWORD**, and confirm the password by retyping it in the **CONFIRM PASSWORD** text box.

  If key-based authentication is implemented between the eG agent and the SSH daemon of a Linux guest, then, in the **ADMIN USER** text box, enter the name of the user whose **<USER_HOME_DIR>** (on that Linux guest) contains a **.ssh** directory with the *public key file* named **authorized_keys**. The **ADMIN PASSWORD** in this case will be the *passphrase* of the *public key*; the default *public key file* that is bundled with the eG agent takes the password *eginnovations*. Specify this as the **ADMIN PASSWORD** if you are using the default private/public key pair that is bundled with the eG agent to implement key-based authentication. On the other hand, if you are generating a new public/private key pair for this purpose, then use the *passphrase* that you provide while generating the pair. For the detailed procedure on *Implementing Key-based Authentication* refer to Section 2.6 of this document.

- **If the guests 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 2.4.1.1.1 of this document.

- **If the INSIDE VIEW USING flag is set to 'eG VM Agent (Windows)'** - In this case, the inside view can be obtained without domain administrator privileges. Therefore, set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.

13. **REPORT BY USER** - While monitoring a VirtualBox, the **REPORT BY USER** flag is set to **Yes** by default, indicating that by default, the guest operating systems on the VirtualBox are identified using the login of the user who is accessing the guest OS. In other words, this test will, by default, report measures for every *username_on_virtualmachinename*. If this flag is set to **No**, then the guests will be identified using the host name of the guest OS. In this case, the test will report measures for every *virtualmachinename*.

	14. **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.		
Outputs of the test	One set of results for every *Linux virtual_guest:network_interface* combination		
Measurements made by the test	**Measurement**	**Measurement Unit**	**Interpretation**
	Incoming traffic: Indicates the rate of incoming traffic.	Pkts/Sec	An increase in traffic to the guest can indicate an increase in accesses to the guest (from users or from other applications) or that the guest is under an attack of some form.
	Outgoing traffic: Represents the rate of outgoing traffic.	Pkts/Sec	An increase in traffic from the guest can indicate an increase in accesses to the guest (from users or from other applications).

2.5.8 Tcp - VM Test

This test tracks various statistics pertaining to TCP connections to and from each desktop of an Oracle VirtualBox. The details of the test are provided below:

Purpose	To measure statistics pertaining to the TCP layer of a desktop
Target of the test	An Oracle VirtualBox
Agent deploying the test	An internal/remote agent

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 –** Refers to the port used by the specified **HOST**. 4. **ORACLE HYPERVISOR USER** - Specify the name of the user who has the right to access the VirtualBox via SSH. 5. **ORACLE HYPERVISOR PASSWORD** - Provide the password of the **ORACLE HYPERVISOR USER**. 6. **CONFIRM PASSWORD** - Confirm the password by retyping it here. 7. **SUDOCMD** - This test executes certain privileged VDA (Virtual Desktop Access) commands to pull out the desired metrics from the VirtualBox. To enable the test to run these commands, you first need to install a **sudo** package on the VirtualBox host. The procedure for installing this package is detailed in Section 1.4.1.2 of this document. Once the package is installed, you need to specify the full path to the install directory of the **sudo** package in the **SUDOCMD** text box. 8. **IGNORE VMS INSIDE VIEW** - Administrators of some high security virtualized 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 an Oracle VirtualBox 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. **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 with Windows VMs in particular, 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 (Windows)** 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 1.5 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*.

12. **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. In order to obtain a remote connection, the test must be configured with user privileges that allow remote communication with the virtual guests. 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 guests do not belong to any domain (as in the case of Linux/Solaris guests)** : In this case, specify "none" in the **DOMAIN** field, and specify a local administrator account name in the **ADMIN USER** below.

      Prior to this, you need to ensure that the same local administrator account is available or is explicitly created on each of the virtual machines to be monitored. Then, proceed to provide the password of the **ADMIN USER** against **ADMIN PASSWORD**, and confirm the password by retyping it in the **CONFIRM PASSWORD** text box.

	If key-based authentication is implemented between the eG agent and the SSH daemon of a Linux guest, then, in the **ADMIN USER** text box, enter the name of the user whose **<USER_HOME_DIR>** (on that Linux guest) contains a **.ssh** directory with the *public key file* named **authorized_keys**. The **ADMIN PASSWORD** in this case will be the *passphrase* of the *public key*; the default *public key file* that is bundled with the eG agent takes the password *eginnovations*. Specify this as the **ADMIN PASSWORD** if you are using the default private/public key pair that is bundled with the eG agent to implement key-based authentication. On the other hand, if you are generating a new public/private key pair for this purpose, then use the *passphrase* that you provide while generating the pair. For the detailed procedure on *Implementing Key-based Authentication* refer to Section 2.6 of this document. ➢ **If the guests 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 2.4.1.1.1 of this document. ➢ **If the INSIDE VIEW USING flag is set to 'eG VM Agent (Windows)'** - In this case, the inside view can be obtained without domain administrator privileges. Therefore, set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*. 13. **REPORT BY USER** - While monitoring a VirtualBox, the **REPORT BY USER** flag is set to **Yes** by default, indicating that by default, the guest operating systems on the VirtualBox are identified using the login of the user who is accessing the guest OS. In other words, this test will, by default, report measures for every *username_on_virtualmachinename*. If this flag is set to **No**, then the guests will be identified using the host name of the guest OS. In this case, the test will report measures for every *virtualmachinename*. 14. **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.		
Outputs of the test	One set of results for each powered-on desktop/logged-in user on the Oracle VirtualBox monitored		
Measurements made by the	**Measurement**	**Measurement Unit**	**Interpretation**

test	**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 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.

2.5.9 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 desktop, and particularly monitors retransmissions.

Purpose	Monitors the TCP protocol traffic to and from each desktop of an Oracle VirtualBox, and particularly measures the percentage of retransmission
Target of the	An Oracle VirtualBox

test	
Agent deploying the test	An internal/remote agent

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 –** Refers to the port used by the specified **HOST**. 4. **ORACLE HYPERVISOR USER** - Specify the name of the user who has the right to access the VirtualBox via SSH. 5. **ORACLE HYPERVISOR PASSWORD** - Provide the password of the **ORACLE HYPERVISOR USER**. 6. **CONFIRM PASSWORD** - Confirm the password by retyping it here. 7. **SUDOCMD** - This test executes certain privileged VDA (Virtual Desktop Access) commands to pull out the desired metrics from the VirtualBox. To enable the test to run these commands, you first need to install a **sudo** package on the VirtualBox host. The procedure for installing this package is detailed in Section 1.4.1.2 of this document. Once the package is installed, you need to specify the full path to the install directory of the **sudo** package in the **SUDOCMD** text box. 8. **IGNORE VMS INSIDE VIEW** - Administrators of some high security virtualized 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 an Oracle VirtualBox 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. **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 with Windows VMs in particular, 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 (Windows)** 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 1.5 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*.

12. **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. In order to obtain a remote connection, the test must be configured with user privileges that allow remote communication with the virtual guests. 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 guests do not belong to any domain (as in the case of Linux/Solaris guests)** : In this case, specify "none" in the **DOMAIN** field, and specify a local administrator account name in the **ADMIN USER** below.

      Prior to this, you need to ensure that the same local administrator account is available or is explicitly created on each of the virtual machines to be monitored. Then, proceed to provide the password of the **ADMIN USER** against **ADMIN PASSWORD**, and confirm the password by retyping it in the **CONFIRM PASSWORD** text box.

If key-based authentication is implemented between the eG agent and the SSH daemon of a Linux guest, then, in the **ADMIN USER** text box, enter the name of the user whose **<USER_HOME_DIR>** (on that Linux guest) contains a **.ssh** directory with the *public key file* named **authorized_keys**. The **ADMIN PASSWORD** in this case will be the *passphrase* of the *public key*; the default *public key file* that is bundled with the eG agent takes the password *eginnovations*. Specify this as the **ADMIN PASSWORD** if you are using the default private/public key pair that is bundled with the eG agent to implement key-based authentication. On the other hand, if you are generating a new public/private key pair for this purpose, then use the *passphrase* that you provide while generating the pair. For the detailed procedure on *Implementing Key-based Authentication* refer to Section 2.6 of this document.

- **If the guests 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 2.4.1.1.1 of this document.
- **If the INSIDE VIEW USING flag is set to 'eG VM Agent (Windows)'** - In this case, the inside view can be obtained without domain administrator privileges. Therefore, set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.

13. **REPORT BY USER** - While monitoring a VirtualBox, the **REPORT BY USER** flag is set to **Yes** by default, indicating that by default, the guest operating systems on the VirtualBox are identified using the login of the user who is accessing the guest OS. In other words, this test will, by default, report measures for every *username_on_virtualmachinename*. If this flag is set to **No**, then the guests will be identified using the host name of the guest OS. In this case, the test will report measures for every *virtualmachinename*.
14. **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.

	15. **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.		
Outputs of the test	One set of results for each powered-on desktop/currently logged-in user on the Oracle VirtualBox monitored		
Measurements made by the test	**Measurement**	**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	
	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.

2.5.10 Handles Usage - VM Test

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

Purpose	Monitors and tracks the handles opened by processes running in a target Windows desktop
Target of the test	An Oracle VirtualBox
Agent deploying the test	An internal/remote agent

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 –** Refers to the port used by the specified **HOST**. 4. **ORACLE HYPERVISOR USER** - Specify the name of the user who has the right to access the VirtualBox via SSH. 5. **ORACLE HYPERVISOR PASSWORD** - Provide the password of the **ORACLE HYPERVISOR USER**. 6. **CONFIRM PASSWORD** - Confirm the password by retyping it here. 7. **SUDOCMD** - This test executes certain privileged VDA (Virtual Desktop Access) commands to pull out the desired metrics from the VirtualBox. To enable the test to run these commands, you first need to install a **sudo** package on the VirtualBox host. The procedure for installing this package is detailed in Section 1.4.1.2 of this document. Once the package is installed, you need to specify the full path to the install directory of the **sudo** package in the **SUDOCMD** text box. 8. **IGNORE VMS INSIDE VIEW** - Administrators of some high security virtualized 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 an Oracle VirtualBox 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. **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 with Windows VMs in particular, 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 (Windows)** 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 1.5 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*.

12. **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. In order to obtain a remote connection, the test must be configured with user privileges that allow remote communication with the virtual guests. 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 guests do not belong to any domain (as in the case of Linux/Solaris guests)** : In this case, specify "none" in the **DOMAIN** field, and specify a local administrator account name in the **ADMIN USER** below.

      Prior to this, you need to ensure that the same local administrator account is available or is explicitly created on each of the virtual machines to be monitored. Then, proceed to provide the password of the **ADMIN USER** against **ADMIN PASSWORD**, and confirm the password by retyping it in the **CONFIRM PASSWORD** text box.

If key-based authentication is implemented between the eG agent and the SSH daemon of a Linux guest, then, in the **ADMIN USER** text box, enter the name of the user whose **<USER_HOME_DIR>** (on that Linux guest) contains a **.ssh** directory with the *public key file* named **authorized_keys**. The **ADMIN PASSWORD** in this case will be the *passphrase* of the *public key*; the default *public key file* that is bundled with the eG agent takes the password *eginnovations*. Specify this as the **ADMIN PASSWORD** if you are using the default private/public key pair that is bundled with the eG agent to implement key-based authentication. On the other hand, if you are generating a new public/private key pair for this purpose, then use the *passphrase* that you provide while generating the pair. For the detailed procedure on *Implementing Key-based Authentication* refer to Section 2.6 of this document.

- **If the guests 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 2.4.1.1.1 of this document.
- **If the INSIDE VIEW USING flag is set to 'eG VM Agent (Windows)'** - In this case, the inside view can be obtained without domain administrator privileges. Therefore, set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.

13. **REPORT BY USER** - While monitoring a VirtualBox, the **REPORT BY USER** flag is set to **Yes** by default, indicating that by default, the guest operating systems on the VirtualBox are identified using the login of the user who is accessing the guest OS. In other words, this test will, by default, report measures for every *username_on_virtualmachinename*. If this flag is set to **No**, then the guests will be identified using the host name of the guest OS. In this case, the test will report measures for every *virtualmachinename*.
14. **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.

	15. **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 each powered-on desktop/currently logged-in user on the Oracle VirtualBox monitored

Measurements made by the test	Measurement	Measurement Unit	Interpretation
	Handles used by processes: 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.
	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 2.15: 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 2.16: The detailed diagnosis of the Processes using handles above limit in VM measure

2.5.11 Windows Services - VM Test

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

Purpose	Tracks the status (whether running or have stopped) of services executing on Windows desktops
Target of the test	An Oracle VirtualBox
Agent deploying the test	An internal/remote agent

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 –** Refers to the port used by the specified **HOST**. 4. **ORACLE HYPERVISOR USER** - Specify the name of the user who has the right to access the VirtualBox via SSH. 5. **ORACLE HYPERVISOR PASSWORD** - Provide the password of the **ORACLE HYPERVISOR USER**. 6. **CONFIRM PASSWORD** - Confirm the password by retyping it here. 7. **SUDOCMD** - This test executes certain privileged VDA (Virtual Desktop Access) commands to pull out the desired metrics from the VirtualBox. To enable the test to run these commands, you first need to install a **sudo** package on the VirtualBox host. The procedure for installing this package is detailed in Section 1.4.1.2 of this document. Once the package is installed, you need to specify the full path to the install directory of the **sudo** package in the **SUDOCMD** text box. 8. **IGNORE VMS INSIDE VIEW** - Administrators of some high security virtualized 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 an Oracle VirtualBox 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. **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 with Windows VMs in particular, 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 (Windows)** 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 1.5 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*.

12. **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. In order to obtain a remote connection, the test must be configured with user privileges that allow remote communication with the virtual guests. 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 guests do not belong to any domain (as in the case of Linux/Solaris guests)** : In this case, specify "none" in the **DOMAIN** field, and specify a local administrator account name in the **ADMIN USER** below.

      Prior to this, you need to ensure that the same local administrator account is available or is explicitly created on each of the virtual machines to be monitored. Then, proceed to provide the password of the **ADMIN USER** against **ADMIN PASSWORD**, and confirm the password by retyping it in the **CONFIRM PASSWORD** text box.

If key-based authentication is implemented between the eG agent and the SSH daemon of a Linux guest, then, in the **ADMIN USER** text box, enter the name of the user whose **<USER_HOME_DIR>** (on that Linux guest) contains a **.ssh** directory with the *public key file* named **authorized_keys**. The **ADMIN PASSWORD** in this case will be the *passphrase* of the *public key*; the default *public key file* that is bundled with the eG agent takes the password *eginnovations*. Specify this as the **ADMIN PASSWORD** if you are using the default private/public key pair that is bundled with the eG agent to implement key-based authentication. On the other hand, if you are generating a new public/private key pair for this purpose, then use the *passphrase* that you provide while generating the pair. For the detailed procedure on *Implementing Key-based Authentication* refer to Section 2.6 of this document.

- **If the guests 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 2.4.1.1.1 of this document.
- **If the INSIDE VIEW USING flag is set to 'eG VM Agent (Windows)'** - In this case, the inside view can be obtained without domain administrator privileges. Therefore, set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.

13. **REPORT BY USER** - While monitoring a VirtualBox, the **REPORT BY USER** flag is set to **Yes** by default, indicating that by default, the guest operating systems on the VirtualBox are identified using the login of the user who is accessing the guest OS. In other words, this test will, by default, report measures for every *username_on_virtualmachinename*. If this flag is set to **No**, then the guests will be identified using the host name of the guest OS. In this case, the test will report measures for every *virtualmachinename*.
14. **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.

<table>
<tr><td></td><td colspan="3">15. IGNORESERVICES - 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.
16. 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.
17. 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.</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for each powered-on desktop/currently logged-in user on the Oracle VirtualBox monitored</td></tr>
<tr><td rowspan="3">Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
<tr><td>New automatic services started:
Indicates the number of Windows services with startup type as automatic, which were running in the last measurement period.</td><td>Number</td><td>The detailed diagnosis of this measure lists the services (with startup type as automcatic) that are running.</td></tr>
<tr><td>New automatic services stopped:
Indicates the number of Windows services with startup type as automatic, which were not running in the last measurement period.</td><td>Number</td><td>To know which services stopped, use the detailed diagnosis of this measure (if enabled).</td></tr>
</table>

	New manual services started: Indicates the number of Windows services with startup type as *manual*, which were running in the last measurement period.	Number	Use the detailed diagnosis of this measure to identify the *manual* services that are running.
	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.

2.5.12 Memory Usage - VM Test

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

Purpose	Reports statistics related to the usage of the physical memory of the desktops
Target of the test	An Oracle VirtualBox
Agent deploying the test	An internal/remote agent

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 –** Refers to the port used by the specified **HOST**. 4. **ORACLE HYPERVISOR USER** - Specify the name of the user who has the right to access the VirtualBox via SSH. 5. **ORACLE HYPERVISOR PASSWORD** - Provide the password of the **ORACLE HYPERVISOR USER**. 6. **CONFIRM PASSWORD** - Confirm the password by retyping it here. 7. **SUDOCMD** - This test executes certain privileged VDA (Virtual Desktop Access) commands to pull out the desired metrics from the VirtualBox. To enable the test to run these commands, you first need to install a **sudo** package on the VirtualBox host. The procedure for installing this package is detailed in Section 1.4.1.2 of this document. Once the package is installed, you need to specify the full path to the install directory of the **sudo** package in the **SUDOCMD** text box. 8. **IGNORE VMS INSIDE VIEW** - Administrators of some high security virtualized 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 an Oracle VirtualBox 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. **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 with Windows VMs in particular, 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 (Windows)** 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 1.5 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*.

12. **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. In order to obtain a remote connection, the test must be configured with user privileges that allow remote communication with the virtual guests. 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 guests do not belong to any domain (as in the case of Linux/Solaris guests)** : In this case, specify "none" in the **DOMAIN** field, and specify a local administrator account name in the **ADMIN USER** below.

      Prior to this, you need to ensure that the same local administrator account is available or is explicitly created on each of the virtual machines to be monitored. Then, proceed to provide the password of the **ADMIN USER** against **ADMIN PASSWORD**, and confirm the password by retyping it in the **CONFIRM PASSWORD** text box.

If key-based authentication is implemented between the eG agent and the SSH daemon of a Linux guest, then, in the **ADMIN USER** text box, enter the name of the user whose **<USER_HOME_DIR>** (on that Linux guest) contains a **.ssh** directory with the *public key file* named **authorized_keys**. The **ADMIN PASSWORD** in this case will be the *passphrase* of the *public key*; the default *public key file* that is bundled with the eG agent takes the password *eginnovations*. Specify this as the **ADMIN PASSWORD** if you are using the default private/public key pair that is bundled with the eG agent to implement key-based authentication. On the other hand, if you are generating a new public/private key pair for this purpose, then use the *passphrase* that you provide while generating the pair. For the detailed procedure on *Implementing Key-based Authentication* refer to Section 2.6 of this document.

- **If the guests 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 2.4.1.1.1 of this document.
- **If the INSIDE VIEW USING flag is set to 'eG VM Agent (Windows)'** - In this case, the inside view can be obtained without domain administrator privileges. Therefore, set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.

13. **REPORT BY USER** - While monitoring a VirtualBox, the **REPORT BY USER** flag is set to **Yes** by default, indicating that by default, the guest operating systems on the VirtualBox are identified using the login of the user who is accessing the guest OS. In other words, this test will, by default, report measures for every *username_on_virtualmachinename*. If this flag is set to **No**, then the guests will be identified using the host name of the guest OS. In this case, the test will report measures for every *virtualmachinename*.
14. **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.

	15. **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 each powered-on desktop/currently logged-in user on the VirtualBox monitored		
Measurements made by the test	**Measurement**	**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.

	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 memory* is *Free physical memory*. Typically, *Available physical memory* is made up of the Standby List, Free List, and Zeroed 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. 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.**
	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.

1.1.1 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.

Note:

This test reports metrics for Windows VMs only.

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 *Oracle VirtualBox* 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.

Purpose	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
Target of the test	An Oracle VirtualBox
Agent deploying the test	An internal/remote agent

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 –** Refers to the port used by the specified **HOST**. 4. **ORACLE HYPERVISOR USER** - Specify the name of the user who has the right to access the VirtualBox via SSH. 5. **ORACLE HYPERVISOR PASSWORD** - Provide the password of the **ORACLE HYPERVISOR USER**. 6. **CONFIRM PASSWORD** - Confirm the password by retyping it here. 7. **SUDOCMD** - This test executes certain privileged VDA (Virtual Desktop Access) commands to pull out the desired metrics from the VirtualBox. To enable the test to run these commands, you first need to install a **sudo** package on the VirtualBox host. The procedure for installing this package is detailed in Section 1.4.1.2 of this document. Once the package is installed, you need to specify the full path to the install directory of the **sudo** package in the **SUDOCMD** text box. 8. **IGNORE VMS INSIDE VIEW** - Administrators of some high security virtualized 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 an Oracle VirtualBox 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. **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 with Windows VMs in particular, 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 (Windows)** 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 1.5 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*.

12. **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. In order to obtain a remote connection, the test must be configured with user privileges that allow remote communication with the virtual guests. 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 guests do not belong to any domain (as in the case of Linux/Solaris guests)** : In this case, specify "none" in the **DOMAIN** field, and specify a local administrator account name in the **ADMIN USER** below.

      Prior to this, you need to ensure that the same local administrator account is available or is explicitly created on each of the virtual machines to be monitored. Then, proceed to provide the password of the **ADMIN USER** against **ADMIN PASSWORD**, and confirm the password by retyping it in the **CONFIRM PASSWORD** text box.

If key-based authentication is implemented between the eG agent and the SSH daemon of a Linux guest, then, in the **ADMIN USER** text box, enter the name of the user whose **<USER_HOME_DIR>** (on that Linux guest) contains a **.ssh** directory with the *public key file* named **authorized_keys**. The **ADMIN PASSWORD** in this case will be the *passphrase* of the *public key*; the default *public key file* that is bundled with the eG agent takes the password *eginnovations*. Specify this as the **ADMIN PASSWORD** if you are using the default private/public key pair that is bundled with the eG agent to implement key-based authentication. On the other hand, if you are generating a new public/private key pair for this purpose, then use the *passphrase* that you provide while generating the pair. For the detailed procedure on *Implementing Key-based Authentication* refer to Section 2.6 of this document.

- **If the guests 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 2.4.1.1.1 of this document.
- **If the INSIDE VIEW USING flag is set to 'eG VM Agent (Windows)'** - In this case, the inside view can be obtained without domain administrator privileges. Therefore, set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.

13. **REPORT BY USER** - While monitoring a VirtualBox, the **REPORT BY USER** flag is set to **Yes** by default, indicating that by default, the guest operating systems on the VirtualBox are identified using the login of the user who is accessing the guest OS. In other words, this test will, by default, report measures for every *username_on_virtualmachinename*. If this flag is set to **No**, then the guests will be identified using the host name of the guest OS. In this case, the test will report measures for every *virtualmachinename*.
14. **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.

Outputs of the test	One set of results for each powered-on Windows desktop/currently logged-in user on the VirtualBox monitored		
Measurements made by the test	Measurement	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.

2.5.13 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.

Purpose	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
Target of the test	An Oracle VirtualBox
Agent deploying the test	An internal/remote agent

Configurable parameters for the test	15. **TEST PERIOD** - How often should the test be executed 16. **HOST** - The host for which the test is to be configured 17. **PORT –** Refers to the port used by the specified **HOST**. 18. **ORACLE HYPERVISOR USER** - Specify the name of the user who has the right to access the VirtualBox via SSH. 19. **ORACLE HYPERVISOR PASSWORD** - Provide the password of the **ORACLE HYPERVISOR USER**. 20. **CONFIRM PASSWORD** - Confirm the password by retyping it here. 21. **SUDOCMD** - This test executes certain privileged VDA (Virtual Desktop Access) commands to pull out the desired metrics from the VirtualBox. To enable the test to run these commands, you first need to install a **sudo** package on the VirtualBox host. The procedure for installing this package is detailed in Section 1.4.1.2 of this document. Once the package is installed, you need to specify the full path to the install directory of the **sudo** package in the **SUDOCMD** text box. 22. **IGNORE VMS INSIDE VIEW** - Administrators of some high security virtualized 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 an Oracle VirtualBox 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. 23. **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.

24. **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.

25. **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 with Windows VMs in particular, 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 (Windows)** 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 1.5 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*.

26. **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. In order to obtain a remote connection, the test must be configured with user privileges that allow remote communication with the virtual guests. 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 guests do not belong to any domain (as in the case of Linux/Solaris guests)** : In this case, specify "none" in the **DOMAIN** field, and specify a local administrator account name in the **ADMIN USER** below.

      Prior to this, you need to ensure that the same local administrator account is available or is explicitly created on each of the virtual machines to be monitored. Then, proceed to provide the password of the **ADMIN USER** against **ADMIN PASSWORD**, and confirm the password by retyping it in the **CONFIRM PASSWORD** text box.

If key-based authentication is implemented between the eG agent and the SSH daemon of a Linux guest, then, in the **ADMIN USER** text box, enter the name of the user whose **<USER_HOME_DIR>** (on that Linux guest) contains a **.ssh** directory with the *public key file* named **authorized_keys**. The **ADMIN PASSWORD** in this case will be the *passphrase* of the *public key*; the default *public key file* that is bundled with the eG agent takes the password *eginnovations*. Specify this as the **ADMIN PASSWORD** if you are using the default private/public key pair that is bundled with the eG agent to implement key-based authentication. On the other hand, if you are generating a new public/private key pair for this purpose, then use the *passphrase* that you provide while generating the pair. For the detailed procedure on *Implementing Key-based Authentication* refer to Section 2.6 of this document.

- **If the guests 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 2.4.1.1.1 of this document.
- **If the INSIDE VIEW USING flag is set to 'eG VM Agent (Windows)'** - In this case, the inside view can be obtained without domain administrator privileges. Therefore, set the **DOMAIN**, **ADMIN USER**, and **ADMIN PASSWORD** parameters to *none*.

27. **REPORT BY USER** - While monitoring a VirtualBox, the **REPORT BY USER** flag is set to **Yes** by default, indicating that by default, the guest operating systems on the VirtualBox are identified using the login of the user who is accessing the guest OS. In other words, this test will, by default, report measures for every *username_on_virtualmachinename*. If this flag is set to **No**, then the guests will be identified using the host name of the guest OS. In this case, the test will report measures for every *virtualmachinename*.
28. **REPORT POWERED OS** - **This flag becomes relevant only if the REPORT BY USER flag is set to 'Yes'.**

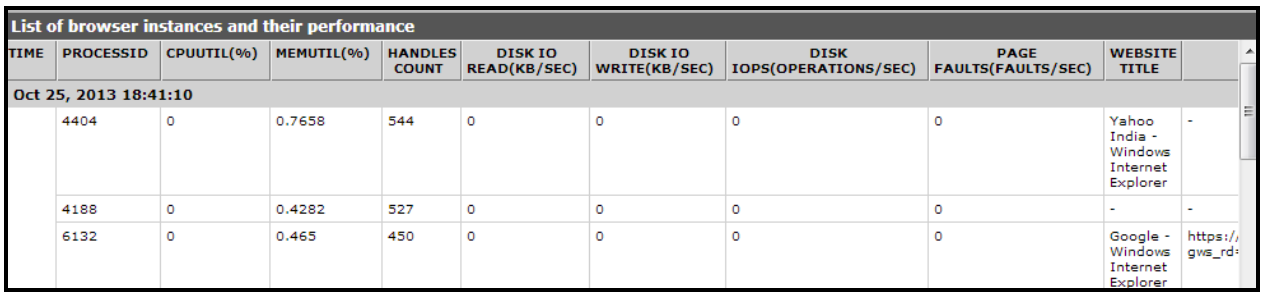
    If the **REPORT POWERED OS** flag is set to **Yes** (which is the default setting), then this test will report measures for even those VMs that do not have any users logged in currently. Such guests will be identified by their *virtual machine name* and not by the *username_on_virtualmachinename*. On 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.
29. **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**.

	30. **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 each browser used by every powered-on Windows desktop/currently logged-in user to the Windows desktop on the VirtualBox monitored

Measurements made by the test	Measurement	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.
	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 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 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.



List of browser instances and their performance

TIME	PROCESSID	CPUUTIL(%)	MEMUTIL(%)	HANDLES COUNT	DISK IO READ(KB/SEC)	DISK IO WRITE(KB/SEC)	DISK IOPS(OPERATIONS/SEC)	PAGE FAULTS(FAULTS/SEC)	WEBSITE TITLE	
Oct 25, 2013 18:41:10										
	4404	0	0.7658	544	0	0	0	0	Yahoo India - Windows Internet Explorer	-
	4188	0	0.4282	527	0	0	0	0	-	-
	6132	0	0.465	450	0	0	0	0	Google - Windows Internet Explorer	https://gws_rd=

Figure 2.17: 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.

Details of websites that are last browsed

TIME	WEBSITE TITLE	WEBSITE URL
Oct 25, 2013 18:41:10		
	Google - Windows Internet Explorer	https://www.google.co.in/?gws_rd=cr&ei=DBhmUpWMC4nYrQez3IDYBQ
	Yahoo India - Windows Internet Explorer	-

Figure 2.18: The detailed diagnosis of the Recent web sites measure

As stated earlier, by default, clicking on the **Virtual Desktop** layer, leads you to a page displaying the current status of the individual desktops that have been configured on the Oracle VirtualBox. If you want to override this default setting - i.e., if you prefer to view the tests mapped to the **Virtual Desktop** layer first, and then proceed to focus on individual desktop 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 **Virtual Desktop** layer is clicked, the list of tests mapped to that layer appears. If you now want the **Desktop view** of Figure 2.10, simply click on **Back** button in the layer model page.

From the desktop view, you can further drill-down to focus on the health of a particular desktop, by clicking on the icon representing the desktop in Figure 2.10. Figure 2.19 then appears displaying all the performance metrics extracted from that virtual desktop 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 desktop. To view the time-of-day variations in a measure, you can view its graph by clicking on that measure in Figure 2.19.

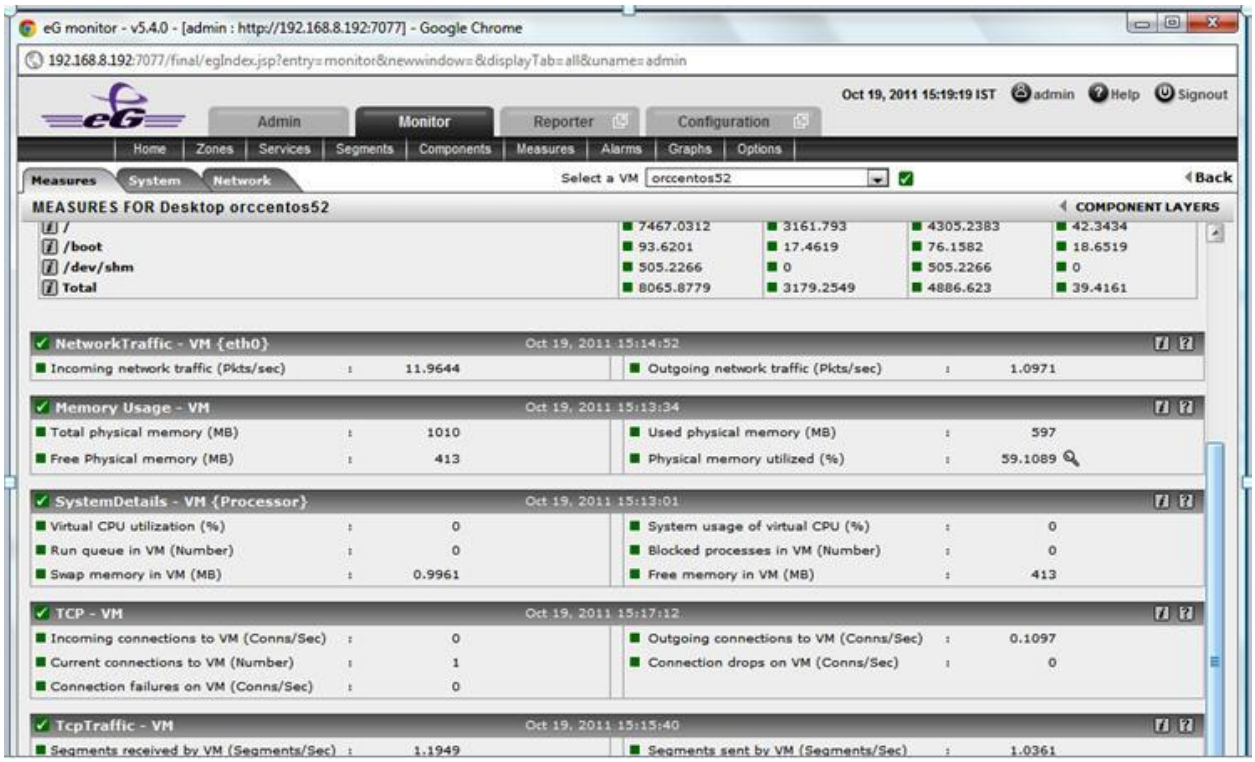


Figure 2.19: The measures pertaining to a particular desktop

You can also view live graphs of pre-configured measures pertaining to the Oracle VirtualBox and the virtual desktops configured on it, by clicking on the **LIVE GRAPHS** link in Figure 2.10. Alternatively, you can click on the icon that appears in the **Tests** panel of the layer model page when the **Virtual Desktop** layer is clicked to view the live graph.

2.6 Troubleshooting the Failure of the eG Remote Agent to Connect to or Report Measures for Linux Guests

By default, the eG agent uses secure shell (SSH) to connect to Linux guests, and collect performance metrics from them. Password Authentication is the default method for SSH connections in eG Enterprise. If the eG agent fails to report measures for a Linux guest or is unable to connect to a guest, it could imply that the Linux guest does not support SSH or that password authentication is not supported by the SSH daemon running on the Linux guest. Under such circumstances, you can perform either of the following:

- Enable Password Authentication in the SSH daemon on the Linux guest; or,
- Implement Key-Based Authentication between eG agent and the SSH daemon of the Linux guest.

If you pick option (1), then follow the steps given below to enable password authentication:

- Login to the Linux guest to be monitored.
- Edit the **sshd_config** file in the **/etc/ssh** directory.

- Check whether the **Password Authentication** flag in the **sshd_config** file is set to **no**. If so, set it to **yes**.
- Then, save the file and restart/signal the SSH daemon (eg., using **kill -1 <sshd_config PID>**).

On the contrary, if you choose to enable key-based authentication [i.e, option (2)], then you will have to generate a public/private key pair. A public/private key pair is available in the **<EG_INSTALL_DIR>\agent\sshkeys** directory (on Windows; on Unix, this will be **/opt/egurkha/agent/sshkeys**) of the eG agent. While the private key is available in the file named **id_rsa**, the public key is contained within the file **authorized_keys**. You now have the option to proceed with the default keys or generate a different key pair. If you decide to go with the keys bundled with the eG agent, do the following:

- To enable key-based authentication, the private key should remain in the **<EG_INSTALL_DIR>\agent\sshkeys** directory (on Windows; on Unix, this will be **/opt/egurkha/agent/sshkeys**), and the public key should be copied to each of the Linux guests to be monitored. To achieve this, first login to the Linux guest to be monitored as the eG user.
- Create a directory named **.ssh** in the **<USER_HOME_DIR>** on the guest operating system, using the command: **mkdir ~/.ssh**.
- Next, copy the **authorized_keys** file from the **<EG_INSTALL_DIR>\agent\sshkeys** directory (on Windows; on Unix, this will be **/opt/egurkha/agent/sshkeys**) on the eG remote agent host to the **<USER_HOME_DIR>/.ssh** directory on the Linux guest.
- Make sure that the permission of the **.ssh** directory and the **authorized_keys** file is **700**.
- Finally, on the eG manager host, edit the **<EG_INSTALL_DIR>\manager\config\eg_tests.ini** file. Against the **EgJavaSSHKeyFile** parameter, enter: **agent/sshkeys/id_rsa.pub**, and save the file.

On the other hand, if you want to generate a new key pair, then do the following:

- Login to any Linux host in your environment (even a Linux VM) as an eG user.
- From the **<USER_HOME_DIR>**, execute the command: **ssh-keygen -t rsa**. Upon executing the command, you will be requested to specify the full path to the file to which the key is to be saved. By default, a directory named **.ssh** will be created in the **<USER_HOME_DIR>**, to which the key pair will be saved. To go with the default location, simply press **Enter**.

```
Generating public/private rsa key pair.
Enter file in which to save the key (/home/egurkha/.ssh/id_rsa):
```

- Next, you will be prompted to provide a pass phrase. Provide any pass phrase of your choice.

```
Enter passphrase (empty for no passphrase): eginnovations
Enter same passphrase again: eginnovations
```

- If the key pair is created successfully, then the following messages will appear:

```
Your identification has been saved in /hom
e/egurkha/.ssh/id_rsa.
Your public key has been saved in /home/egurkha/.ssh/id_rsa.pub.
The key fingerprint is:
09:f4:02:3f:7d:00:4a:b4:6d:b9:2f:c1:cb:cf:0e:e1 dclements@sde4.freshwater.com
```

- The messages indicate that the private key has been saved to a file named **id_rsa** in the **<USER_HOME_DIR>/.ssh**, and the public key has been saved to a file named **id_rsa.pub** in the same directory. Now, to enable key-based authentication, login to the Linux guest to be monitored as the eG user.

- Create a directory named **.ssh** in the **<USER_HOME_DIR>** on the guest operating system, using the command: **mkdir ~/.ssh**.
- Next, copy the **id_rsa.pub** file from the **<USER_HOME_DIR>/.ssh** directory on the Linux host to the **<USER_HOME_DIR>/.ssh** directory on the Linux guest.
- Ensure that the **id_rsa.pub** file on the Linux guest is renamed as **authorized_keys**.
- Repeat this procedure on every Linux guest to be monitored.
- Then, lock the file permissions down to prevent other users from being able to read the key pair data, using the following commands:

  chmod go-w ~/
  chmod 700 ~/.ssh
  chmod go-rwx ~/.ssh/*

- Finally, on the eG manager host, edit the **<EG_INSTALL_DIR>\manager\config\eg_tests.ini** file. Against the **EgJavaSSHKeyFile** parameter, enter: **agent/sshkeys/id_rsa.pub**, and save the file.

Instead of choosing between the authentication modes (Password or Key-based), you can also disable the usage of the Java SSH client, and use **plink** to connect to Linux guests. To achieve this, follow the steps given below:

- Edit the **eg_tests.ini** file in the **/opt/egurkha/manager/config** directory (on Unix; on Windows, this will be **<EG_INSTALL_DIR>\manager\config** directory).
- Set the **JavaSSHForVm** flag in the **[AGENT_SETTINGS]** section of the file to **false**; by default, this is set to **true** indicating that the eG agent uses Java SSH by default. By setting the flag to **false**, you can ensure that the eG agent does not use Java SSH, and instead uses the **plink** command to connect to Linux guests.
- The **plink** command exists in the **<EG_INSTALL_DIR>\lib\vmgfiles** directory (on Windows; on Unix, this will be **/opt/egurkha/lib/vmgfiles**) of the eG agent. To use the **plink** command, you need to explicitly configure the SSH keys, so that the eG agent is able to communicate with the Linux guests using SSH. To do this, follow the steps given below:
  - Go to the command prompt and switch to the directory containing the **plink** command.
  - Then, execute the **plink** command to connect to any of the Linux-based virtual machines on the vSphere host. The syntax for the **plink** command is as follows:

    plink -ssh <user>@<IP_of_target_host> <command>

    For example, assume that you want to connect to the virtual machine, **192.168.10.7**, as user **john** with password **john**, to know its hostname. The syntax of the **plink** command in this case will be:

    plink -ssh john@192.168.10.7 hostname, where **hostname** is the command to be executed on the remote host for extracting its hostname.

  - To ensure that you do not connect to an imposter host, **SSH2.x** presents you with a unique host key fingerprint for that host, and requests your confirmation to save the displayed host key to the cache.

    ```
    The server's host key is not cached in the registry. You have no guarantee
    that the server is the computer you think it is.
    The server's rsa2 key fingerprint is:<host key>
    If you trust this host, enter "y" to add the key to PuTTY's cache and carry
    on connecting.
    If you want to carry on connecting just once, without adding the key to the
    cache, enter "n".
    ```

```
If you do not trust this host, press Return to abandon the connection.
Store key in cache? (y/n) y
```

Once you confirm the host key storage and provide the user's password to connect to the virtual guest, this message will not appear during your subsequent attempts to connect to any Linux-based virtual machine on the monitored vSphere/ESX host. In other words, the eG agent will be able to execute tests on all Linux guests on the target ESX host without any interruption. Therefore, press **y** to confirm key storage.

Chapter 3

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 the **Oracle VirtualBox**. 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.