

CONSOLIDATION



**CONSOLIDATING SQL SERVER® 2008
ONTO DELL™ POWEREDGE™ R900 AND
POWEREDGE™ R905 USING
MICROSOFT'S HYPER-V™**

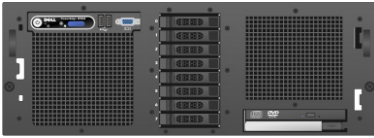
A Principled Technologies report commissioned by Dell

Table of contents

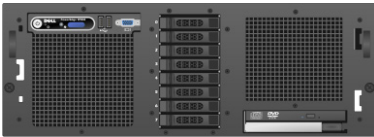
Table of contents	2
Introduction	3
Overview of this companion Guide.....	3
Continuing the migration and consolidation process	4
Creating and configuring your virtual machines	5
Creating a virtual machine	5
Installing Windows Server 2008 in your virtual machine	6
Installing SQL Server 2008 in your virtual machine	8
Moving to your virtual environment	13
Moving your database	13
Transferring logins	15
Summing up	17
Appendix A. Creating the virtual machine and installing the guest OS	18
Creating the virtual machine	18
Adding processors and virtual hard drives to the virtual machine	18
Installing the Windows Server 2008 guest OS	20
Setting IP address, subnet mask, and domain information on your VM	21
Configuring virtual data drives	22
Appendix B. Installing SQL Server 2008 Enterprise Edition.	24
Appendix C. Migrating SQL Server 2008 databases from your physical machine to your virtual machine	28
Appendix D. Transferring Windows logins via SQL Server Management Studio	31
Appendix E. Transferring SQL Server logins via SQL Server Management Studio	33
About Principled Technologies	36

Introduction

Overview of this companion Guide



Dell PowerEdge R900



Dell PowerEdge R905

This Guide is a companion to *How to Guide: Consolidating multiple SQL Server® systems onto Dell™ PowerEdge™ Servers using Microsoft's Hyper-V™*. That Guide provides general virtualization and consolidation concepts, and demonstrates the setup and basic configuration of the Dell™ PowerEdge™ R900 or PowerEdge™ R905 server. Building on that base, this Guide adds the techniques you need to successfully consolidate your Microsoft® SQL Server® 2008 databases from multiple machines onto a single Windows Server® 2008 server system using Microsoft's Hyper-V™. In this Guide, we use the Dell PowerEdge R900 and PowerEdge R905 as target servers.

Principled Technologies® (PT) has performed hands-on testing and research and drawn on real-world experiences to document best practices and help you take advantage of the features of Hyper-V to consolidate multiple SQL Server 2008 instances onto a single physical system.

This document covers the concepts and procedures specific to virtualizing SQL Server 2008 on the Hyper-V platform. It reviews in detail the approach that PT and Dell recommend for a tested and validated configuration of Hyper-V and SQL Server 2008. For general concepts and procedures on setting up your virtualization server with Windows Server 2008 and Hyper-V, see our accompanying Guide, *How to Guide: Consolidating multiple SQL Server® systems onto Dell™ PowerEdge™ Servers using Microsoft's Hyper-V™*. Using this SQL Server 2008 Guide in conjunction with the general Hyper-V overview Guide, you can easily virtualize your SQL Server 2008 environment.

The Dell PowerEdge R900 supports up to 24 logical processors, or cores, and the Dell PowerEdge R905 supports 16 logical processors. The release version of Microsoft Hyper-V supports a maximum of 16 logical processors. See

<http://technet.microsoft.com/en-us/library/cc816844.aspx> for more details on Hyper-V hardware considerations.



NOTE: Microsoft released an update in September 2008 to increase this maximum to 24 logical processors. For more details, and to download this update, visit <http://support.microsoft.com/kb/956710>.

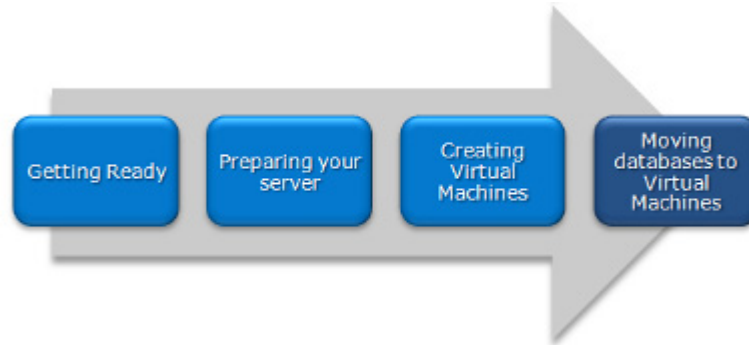
Because we use both the Dell PowerEdge R900 and PowerEdge R905 in this Guide, we refer to the target server as the PowerEdge, and note specifics to each platform where they apply.

Continuing the migration and consolidation process

In the accompanying Guide, *How to Guide: Consolidating multiple SQL Server® systems onto Dell™ PowerEdge™ Servers using Microsoft's Hyper-V™*, we discussed best practices and gave detailed steps on setting up and configuring PowerEdge server with Windows Server 2008 and Hyper-V. We also covered best practices in disk layout as it relates to a SQL Server workload.

We now discuss procedures involved in migrating your SQL Server 2008 databases to a virtualized environment. The steps we list below are a continuation of the process began in the companion Guide. This Guide assumes you have followed the procedures in the previous Guide; the instructions here may not work if you have not followed those steps exactly.

Creating and configuring your virtual machines



Creating a virtual machine

After you have installed the Hyper-V role, the hypervisor will be running and ready for you to create virtual machines. In this section, we provide basic steps for creating new virtual machines, and show how to allocate resources for those machines. For detailed steps, see [Appendix A](#).

1. Open Hyper-V manager under Start | Administrative tools. Connect to your server. Then, choose Action | New | Virtual Machine to start the Virtual Machine creation wizard.
2. Give the virtual machine a name, and provide memory, NIC, and VHD details. Choose the appropriate disk drive to store your new VHD (see Figure 1). In our case, we created a new VHD file on the C: drive of the parent partition.

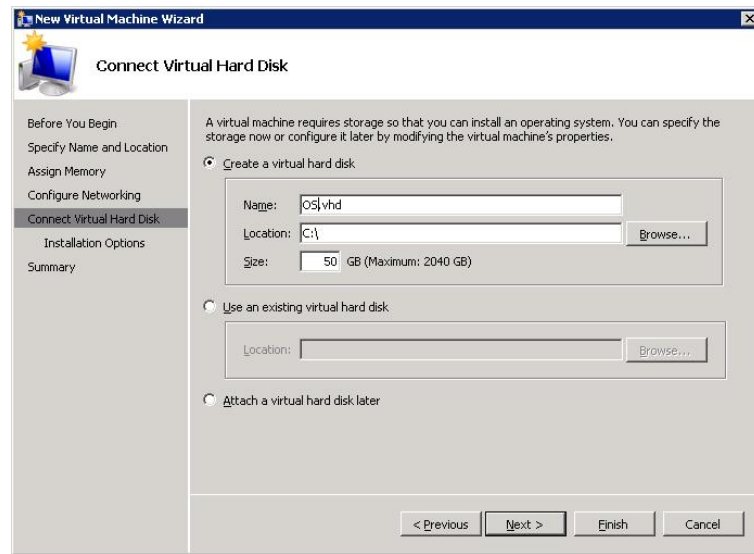


Figure 1. Configuring a Hyper-V virtual machine.

3. Choose Install an operating system later. We cover these steps below in the next section.
4. Finish the wizard, then right-click the virtual machine, choose Settings, and configure multiple processors and other devices as necessary.
5. Finally, create a SCSI controller for this VM. Create additional VHDs, and attach them to this SCSI controller: one for tempdb, one for SQL Server logs, and one for SQL Server user data.

Installing Windows Server 2008 in your virtual machine

To make the virtual machine operable, you will of course need to install an operating system, just as you would on a physical machine. Virtualizing your machine allows you to carry over the same operating system as your physical machine had. In this section, we show how to install Windows Server 2008. For details on both procedures, see [Appendix A](#).



NOTE: Microsoft offers a product called System Center Virtual Machine Manager (SCVMM) that allows you to perform a physical to virtual (P2V) machine migration with just a few clicks. If you have highly customized your configuration and a new install of the operating system and SQL Server would not be appropriate,

consider using SCVMM or another third-party product to perform the P2V conversion.

This section provides an overview of the Windows Server 2008 installation process on a Hyper-V virtual machine. For the most part, this installation on the virtual machine is identical to the typical installation the physical machine. We include approximate wait times for each step. ([Appendix A](#) provides complete, detailed installation instructions.)



NOTE: Plan on at least 60 minutes for installing Windows Server 2008 Enterprise Edition on a Hyper-V virtual machine. Each step below takes at least 1 minute. We provide the approximate amount of time each step takes in parentheses. These times exclude data entry time. The time to install updates—10 minutes in our setup—will increase over time as Microsoft releases additional OS updates.

1. Insert the Microsoft Windows Server 2008 Enterprise Edition DVD into the physical DVD drive of your PowerEdge server.
2. Start Hyper-V manager, and connect to the relevant VM. Attach the physical optical device to the virtual machine using the Media menu.
3. Start the VM, and proceed with the Windows Server 2008 installation steps on your virtual machine, just as you installed Windows Server 2008 on your physical machine.
4. Set the basic location information, and enter your product key, if applicable. Enter any other information the installation software requires. (18 minutes on the PowerEdge R900; 21 minutes on the PowerEdge R905, during which Windows resets the server twice)
5. Set your password on the new virtual machine, enter configuration information, including the IP addresses and domain information, and reboot the system. (3 minutes)
6. Start the Hyper-V Integration Services installation on the virtual machine by choosing Action on the VM menu | Insert Integration Services Setup Disk. Proceed through the Integration Services installation, and reboot the VM if necessary. (5 minutes)

7. Download and install updates for the OS. (10 minutes in our case; download times will vary)
8. Configure your disks inside the VM operating system. Follow the same procedures you used when configuring your physical disks, except that these disks are virtual disks, connected to the VHD files you created earlier. (3 minutes per disk, though times vary a bit)
9. Restart the virtual machine after updates complete installation. (3 minutes)

Installing SQL Server 2008 in your virtual machine

To migrate your databases from physical machines to virtual machines, you must have installed and properly configured the appropriate version of SQL Server on them. We cover a new installation of SQL Server 2008 in this section. We include approximate wait times for each step. [Appendix B](#) provides detailed installation instructions.



NOTE: Plan on at least 30 minutes for installing SQL Server 2008 Enterprise Edition on the VM. Each step below takes at least 1 minute. We provide the approximate amount of time each step takes in parentheses. These times exclude data entry time.

1. Insert the SQL Server 2008 Enterprise Edition DVD into the physical DVD drive.
2. Start Hyper-V manager, and connect to the relevant VM. Attach the physical optical device to the virtual machine using the Media menu.
3. If the DVD does not autoplay on your VM, you may have to navigate to the Servers folder on the DVD and double-click setup. Click Installation, then click New SQL Server stand-alone installation or add features to an existing installation.
4. Install prerequisites (see Figure 2) and .NET 3.5, if necessary. The installer may prompt you to install a hotfix as well. Restart the VM as prompted. (10 minutes)

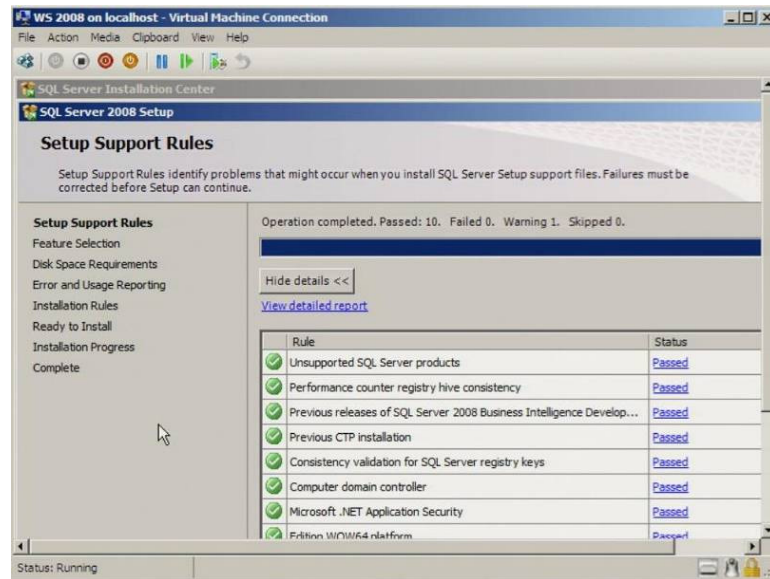


Figure 2. Installing SQL Server 2008 prerequisites.

5. For a basic install, click Database Engine Services, Client Tools connectivity, and Management Tools (Basic and Complete). If you require other components, you can choose to install them here (see Figure 3). (less than 1 minute)

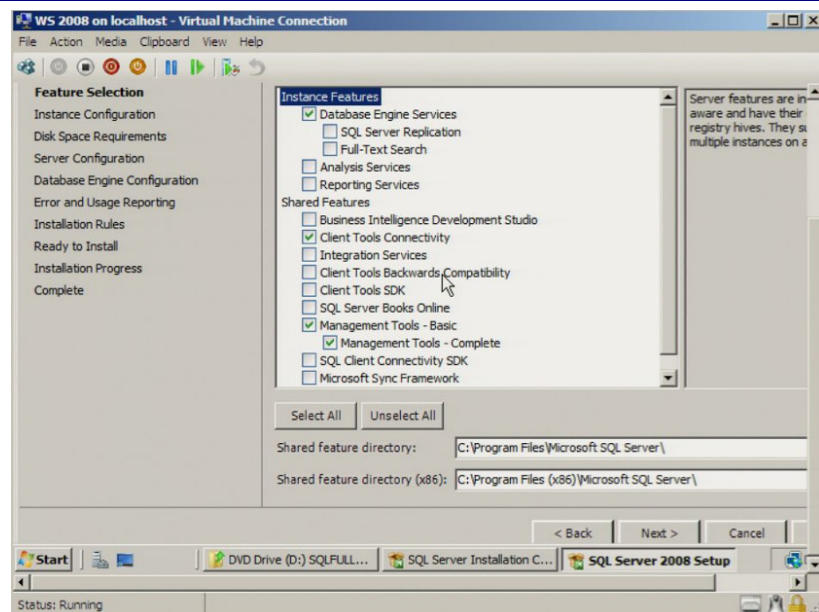


Figure 3. Selecting SQL Server 2008 features.

6. On the following screens, fill in the account information and data file locations. Be sure and point your default data, log, and tempdb locations to the appropriate places (see Figure 4). At the confirmation screen, click Next.



Figure 4. Setting file locations.

7. At the Ready to Install screen, review the summary, and complete the installation. (8 minutes)

When the installation process completes, check the Microsoft Downloads Center (www.microsoft.com/downloads) for the latest SQL Server patches, updates, and service packs.



BEST PRACTICE: After you complete the SQL Server installation on the VM, give the SQL Server service account the right to prevent the operating system from paging its memory to disk. SQL Server dynamically allocates and deallocates memory to relieve memory pressure and swapping. However, another process can always request a substantial amount of memory and thus cause the OS to swap some of SQL Server's memory to disk before SQL Server has the chance to react. This setting prevents that.

Windows Server 2008 Enterprise Edition provides a setting that allows SQL Server to retain its data in physical memory instead of swapping the data to virtual memory, the page file, or disk. This is especially important on 32-bit systems using AWE. To enable this setting, give the account running the SQL Server service the right to lock pages in memory. To do so, follow these steps inside the VM:

1. Click Start.
2. Go to Administrative tools | Local Security Policy.
3. Click Local Policies, and then click User Rights Assignment.
4. Scroll down to Lock pages in memory, and double-click (see Figure 5).
5. Click Add User or Group, and enter the SQL Server service account name.

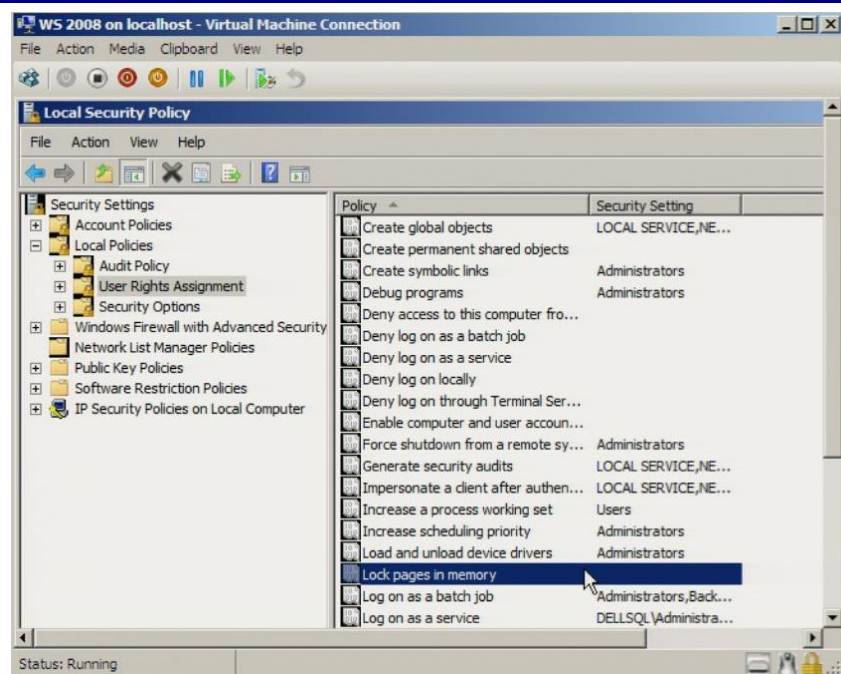


Figure 5. Selecting the Lock pages in memory setting inside your VM.



BEST PRACTICES: In the SQL Server instance memory properties, leave the Minimum server memory and the Maximum server memory at their defaults of 0 and 2147483647, respectively. Doing so allows SQL Server Enterprise Edition to use as much memory as the system makes available. If you must change these

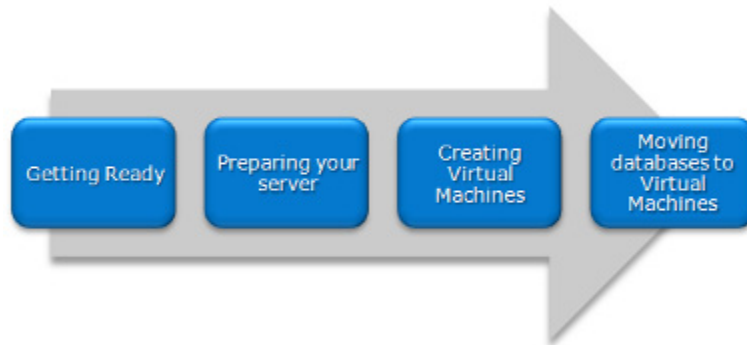
settings, be sure that the sum of the maximum memory settings across all processes is less than the amount of physical RAM available.

Enable instant file initialization. The default behavior is to initialize the storage with zeros whenever SQL Server creates a data file or allocates an extent to grow a data file. Zero-filling storage can be very time-consuming. When you enable instant file initialization, the system does not initialize storage it allocates. Instead, the storage remains uninitialized until SQL Server writes data to it. According to Microsoft, instant file initialization can greatly reduce the performance impact of file growth (see Table 3 at <http://www.microsoft.com/technet/prodtechnol/sql/2005/physdbstor.mspx>).

To enable instant file initialization, you must give the SE_MANAGE_VOLUME_NAME permission to the Windows account under which you run the SQL Server Enterprise Edition service. To do so, allow the account to perform volume maintenance tasks with the following steps:

- 1.** Click Start.
- 2.** Go to Administrative tools | Local Security Policy.
- 3.** Click User Rights assignment.
- 4.** Scroll down to Perform volume maintenance tasks, and double-click.
- 5.** Click Add User or Group, and enter the SQL Server service account name.

Moving to your virtual environment



Moving your database

In this section, we provide an overview of the processes involved in moving your database from a physical server running SQL Server 2008 to a virtual server running the same version. Our assumption for this Guide is that this move entails no upgrades. In other words, because you have virtualized your servers, you will only be moving SQL Server 2008 databases to your virtual machine running SQL Server 2008. Upgrading from an earlier version of SQL Server is outside the scope of this Guide. We provide detailed instructions in [Appendix C](#).

We performed all SQL Server 2008 backup and restore maintenance using SQL Server Management Studio, the main administration interface for SQL Server 2008.



NOTE: You can move databases between instances and versions of SQL Server by various methods, including detach/attach, backup/restore, and the copy database wizard. We chose to use the backup/restore method.

Backing up your SQL Server database on your physical machine

On the original physical server running SQL Server, take the following steps to back up your database. Backup times vary depending on the size of the database.

1. To keep users from issuing updates during the migration process, you can either set the database to read-only mode, or set the access property to SINGLE_USER. Both choices immediately sever all user connections. See [Appendix C](#) for details on setting the database to single-user mode. We chose to set the database to SINGLE_USER mode. Be aware that in SQL

Server Management Studio, it may be necessary to close the Object Explorer Details pane, the Summary pane, and any other parts of the user interface that you may have active. These panes open database connections to the relevant database, and so may cause your SINGLE_USER operation to fail.

2. Perform a full backup of your SQL Server database. See [Appendix C](#) for details.



NOTE: If the time a full backup requires is unacceptable given your migration maintenance schedule, you can make the full backup at a time you previously scheduled, and at this point of the migration only make a differential or transaction log backup. If the transaction log size is small relative to the data file size, this can decrease migration time significantly. However, be certain to keep your backup chain intact if you choose this approach.

Restoring database to the SQL Server instance on your virtual machine

To restore your database to the VM running SQL Server, perform the following steps:

1. Connect to your VM in Hyper-V manager. Open SQL Server Management Studio on the VM, connect to the SQL Server instance, and restore the database (see Figure 6). See [Appendix C](#) for details.

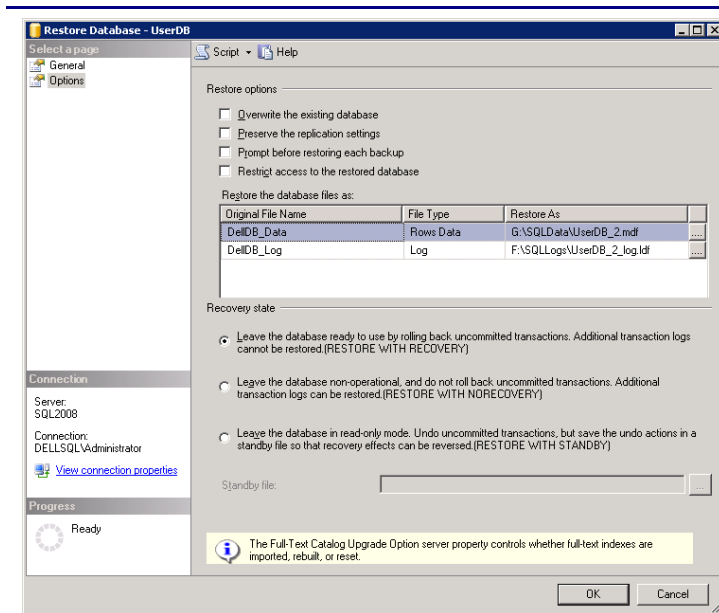


Figure 6. Example SQL Server 2008 restore configuration.



NOTE: It is important at this point to give your new database the same name as your original SQL Server database. Changing the name may break applications that refer to the database by name.

2. While in SQL Server Management Studio, reset the database access property to multi-user. See [Appendix C](#) for details.

Transferring logins

Windows and SQL Server logins

There are two methods of authenticating to SQL Server: Windows logins and SQL Server logins. Windows logins exist in the context of an Active Directory domain, and you can assign those logins rights to SQL Server resources. However, you create and manage SQL Server logins within SQL Server. While the processes for extracting login information and creating the transferred login entities on the consolidation server are very similar for both Windows authenticated logins and SQL Server authenticated logins, you must take some extra steps to ensure a smooth migration for SQL Server logins.



NOTE: Take special care to avoid login issues for users when consolidating your databases, even in a virtual environment. Typically, these issues arise when a user had multiple SQL Server logins on multiple SQL Server instances in the previous environment. In that situation, a user may have used different passwords for different instances. However, depending on which databases you consolidate, the user may have only one login and associated password. You should note other issues in your planning documentation, such as permissions for multiple databases, default database, and language settings.

Below, we describe how to generate Transact SQL (TSQL) scripts that you can use to recreate both Windows authenticated logins and SQL Server authenticated logins on the new virtual SQL Server.

See [Appendix D](#) for details on transferring Windows logins and [Appendix E](#) for details on transferring SQL Server logins.

To transfer logins, take the following steps on the new VM running SQL Server. Note that steps 1 through 5 apply to both Windows logins and SQL Server logins.

1. In SQL Server Management Studio, in the Object Explorer pane, connect to the relevant physical SQL Server and the relevant virtual SQL Server to which you are moving users. Be sure to have the Object Explorer Details tab open (View | Object Explorer Details).
2. Expand the treeview of the source SQL Server, browse to the Security tab, and click the Logins node. In the Object Explorer Details, a list of all logins on the SQL Server appears.
3. If necessary, use the sorting and filtering options in the Object Explorer Details tab, and note which logins you would like to migrate. Select them by clicking and using the standard Windows controls (Ctrl key, Shift key, etc.) to select multiple logins.
4. Right-click the selected logins, and select Script Login As | Create To | New Query Window. If you have not already connected to your new virtual SQL Server, do so now by right-clicking and selecting Change Connection.
5. Execute the script on your virtual SQL Server to create the logins. If you are transferring

Windows-based logins, the process is complete.

For SQL Server logins, the script you execute in step 5 above creates the login, marks it as disabled, assigns it a random password, and does not map the login to the database user. To avoid having a database user who is "orphaned" from the login, use the provided stored procedure called `sp_change_users_login` to view orphaned users and to map the user to the login. We provide an example of this process in [Appendix E](#).

Summing up

As this Guide has explained, the process of consolidating your SQL Server 2008 database servers to a virtual environment on a Dell PowerEdge Server running Hyper-V is relatively straightforward; you can perform a basic installation and move databases in a couple of hours. We still recommend, however, that you first invest ample time in the pre-consolidation planning phase, because doing so can help you avoid potential problems during your consolidation.

Appendix A. Creating the virtual machine and installing the guest OS

Creating the virtual machine

1. Click Start.
2. Expand Administrative Tools, and click Hyper-V Manager.
3. From the Action menu, select Connect to Server.
4. Accept the default of Local Computer, and click OK.
5. From the Action menu, select New | Virtual Machine.
6. On the Before You Begin page, click Next.
7. On the Specify Name and Location page, enter the name for your new virtual machine. Accept the default location to store the Virtual machine, and click Next.
8. On the Assign Memory page, enter the amount of RAM you want to reserve for the virtual machine, in MB.
9. On the Configure Networking, choose Not Connected, and click Next.
10. Set an appropriate size on the Connect Virtual Hard Disk page. Accept the other defaults, and click Next. You will be adding other hard disks later.
11. On the installation Options page, accept the default of Install an operating system later, and click Next.
12. On the Completing the New Virtual Machine Wizard page, click Finish.

Adding processors and virtual hard drives to the virtual machine

Allow at least 10 minutes to add the virtual processors and hard drives.

Add any virtual processors, as well as each of the three non-operating system virtual hard drives to the VM, using the following steps:

1. In the Hyper-V manager window, right-click the virtual machine you just created, and choose Settings

2. By default, Hyper-V creates the VM with one virtual processor. You can raise the number by clicking Processor in the left pane, then setting the number of logical processors.



NOTE: Hyper-V supports four logical processors for Windows Server 2008 64-bit.

3. In the left pane, click Add Hardware.
4. In the right pane, choose SCSI controller, and click Add.
5. In the left pane, choose SCSI controller.
6. In the right pane, select hard drive, and click Add.
7. In the right pane, click New.
8. In the Before You Begin page of the New Virtual Hard Disk Wizard, click Next.
9. On the Choose Disk Type page, choose Fixed size.
10. On the Specify Name and Location page, name the virtual hard disk file, and give the path name for the VHD file. Per best practices, store the virtual disks for the tempdb, logs, and user data on the appropriate drives. We put the relevant VHD file at the root of each drive we configured.
11. On the Configure Disk page, enter the size of the virtual disk. In our case, we used one-quarter the size of the physical RAID for each. Do not copy the contents of another disk. Click Next.
12. On the Completing the New Virtual hard Disk Wizard page, click Finish. (3 minutes per disk, although the time varied from disk to disk.)
13. After adding all three drives, click OK to close the Settings Window.



NOTE: You will notice that the Hyper-V manager created the first virtual disk on a virtual IDE controller. Do not try to change this. As per the Hyper-V documentation, the guest OS must be on a virtual IDE controller.

Installing the Windows Server 2008 guest OS

Allow 60 minutes to complete the installation of the Windows Server 2008 guest OS.

- 1.** In the Hyper-V manager, right-click the virtual machine, and choose connect. You will see a message that the virtual machine is turned off.
- 2.** On the Media menu, choose DVD Drive | Capture D:.
- 3.** Insert the Microsoft Windows Server 2008 Enterprise Edition DVD into the DVD drive. From the Action menu, choose Start. (2 minutes, during which the VM screen is black.)
- 4.** Click inside the VM window to capture the mouse.
- 5.** On the Install Windows Server 2008 screen, accept the defaults of English language, English time and currency format, and US keyboard or input method by clicking Next.
- 6.** Click Install now.
- 7.** On the Type your product key for activation screen, enter your activation key, and click Next.
- 8.** On the Select the edition of Windows that you purchased screen, click Windows Server 2008 Enterprise Edition (Full Installation), check the I have selected the edition of Windows that I purchased box, and click Next.
- 9.** On the Please read the license terms screen, check the I accept the license terms checkbox, and click Next.
- 10.** On the Which type of installation do you want? screen, for Type of install, click Custom (Advanced).
- 11.** On the Where do you want to install Windows? screen, you will only see one choice, because the VM itself defines where the OS should go. Accept the default, and click Next to start the installation. (18 minutes on the Dell PowerEdge R900; 21 minutes on the Dell PowerEdge R905.)
- 12.** When the system indicates that you must change the user's password before you log on for the first time, click OK.

- 13.** Enter a strong password in both the New password and Confirm password fields, and click the arrow.
- 14.** On the Your password has been changed prompt, click OK. Windows then prepares the desktop. (less than 1 minute)
- 15.** From the Action menu for the VM, select Insert Integration Services Setup Disk.
- 16.** When the Autoplay Window appears, click Install Hyper-V Integration Services.
- 17.** On the Upgrade Hyper-V Integration Services dialog box, click OK. Until you upgrade the Integration Services, the mouse, keyboard, video and networking may not work correctly.
- 18.** On the Installation complete dialog box, click No, because you need to stop the VM and select your virtual NIC before proceeding.
- 19.** Turn off the VM. (2 minutes)
- 20.** Now that you have installed the Integration Services, in the Hyper-V manager, right-click the VM, and choose Settings.
- 21.** Click Network adaptor, and select your virtual NIC.
- 22.** Click Apply, and click OK.
- 23.** Right-click the VM, and choose Start.
- 24.** On the prompt to use Ctrl+Alt+Del to log in, click the three-button icon in the upper left corner of the virtual machine's window. If you press the Ctrl+Alt+Del keys, the host operating system will intercept the command.
- 25.** Configure your networking as per the steps in the section Setting IP address, subnet mask, and domain information below.
- 26.** Install Windows updates. Depending on your particular environment, you may need to change the default firewall settings for Windows update to work. (10 minutes; download times vary)
- 27.** Restart the VM. (3 minutes)

Setting IP address, subnet mask, and domain information on your VM

After rebooting the VM, set the static IP addresses and subnet mask on your VM using the following steps:

- 1.** Click Start.
- 2.** Right-click Network, and select Properties.

- 3.** In the Network Sharing Center, click Manage network connections.
- 4.** Right-click the Local Area Connection, and select Properties.
- 5.** In the Local Area Connection Properties dialog, highlight Internet Protocol Version 4 (TCP/IPv4), and click Properties.
- 6.** Click the radio button next to Use the following IP address.
- 7.** Type the IP address you want the server to have, the Subnet mask, and the Preferred DNS server, and click OK.
- 8.** Close the Network Connections window.
- 9.** From the Start menu, right-click Computer, select Properties, and then select Advanced system settings.
- 10.** Select the Computer Name tab.
- 11.** Click Change.
- 12.** Enter the server name.
- 13.** Click the Member of Domain radio button.
- 14.** Enter the domain name.
- 15.** Enter the user name and password when the system prompts you to do so. (less than 1 minute)
- 16.** When the Computer Name/Domain Changes window appears, click OK to close it.
- 17.** Click OK to go past the warning to restart.
- 18.** Click OK to close the System Properties window.
- 19.** Click Restart Now to reboot the server. (3 minutes)

Configuring virtual data drives

Although we configured the data drives at the level of the host operating system, we have to repeat the process for the virtual drives. Configure each of the three non-operating system virtual drives on the virtual machine using the following steps:

- 1.** Click Start.
- 2.** Expand Administrative Tools, and click Computer management.
- 3.** In the left panel, expand Storage, and select Disk Management.
- 4.** If the disk is offline, right-click the disk name, and choose Online. If the Initialize Disk dialog box appears, click OK.

- 5.** Right-click the unallocated disk, and select New Simple Volume.
- 6.** On the Welcome to the New Simple Volume Wizard screen, click Next.
- 7.** Leave the volume size at the default, and click Next.
- 8.** Leave the drive letter at the default, and click Next.
- 9.** On the Format Partition screen, do the following:
 - a. Leave the File System as NTFS.
 - b. Set the Allocation Unit size to 64 K. (Note: Windows Server 2008 Enterprise Edition grays out the option to compress the drive.)
 - c. Fill in a volume name if you want the disk to have one.
 - d. Optionally, select the quick format checkbox.
- 10.** Click Next.
- 11.** On the Completing the New Simple Volume Wizard screen, click Finish.

Appendix B. Installing SQL Server 2008 Enterprise Edition

Install Microsoft SQL Server Enterprise Edition in the relevant virtual machine on the PowerEdge R900 or PowerEdge R905 server by following these steps. We provide approximate times for each group of steps in the first step of that group. Allow 30 minutes for completing the installation.

- 1.** Start Hyper-V manager by clicking Start, then Administrative Tools, then Hyper-V manager.
- 2.** In the Hyper-V manager, right-click the Windows Server 2008 virtual machine, and choose Connect. A message that the virtual machine is turned off appears.
- 3.** On the Media menu, choose DVD Drive | Capture D:.
- 4.** From the Media menu for the virtual machine, make sure that it has captured the DVD drive.
- 5.** Insert the SQL Server 2008 Enterprise Edition DVD into the DVD drive.
- 6.** Start the virtual machine.
- 7.** On the Autoplay Dialog, click Run SETUP.EXE.
Note: You may see two copies of the Autoplay dialog. Be sure to click the one in the VM.
- 8.** On the Microsoft SQL Server 2008 Setup dialog, click OK to install .NET.
- 9.** On the Accept License Terms screen, check I accept the license terms, and click Install. (8 minutes)
- 10.** Exit the .NET installation by clicking Exit.
- 11.** On the Windows Update Standalone Installer dialog, click OK to install the Windows hotfix.
- 12.** When Installation is complete, click Restart Now to restart the VM. (2 minutes)
- 13.** After the restart, log back into the VM.
- 14.** Start the installation process again by running setup.exe on the root directory of the DVD drive.
- 15.** On the SQL Server Installation Center window, click Installation in the left pane.
- 16.** Click New SQL Server stand-alone installation or add features to an existing installation.
- 17.** On the Setup Support Rules screen, click OK. It should have passed all rules.

- 18.** Enter your product key, and click Next.
- 19.** Accept the license terms, and click Next.
- 20.** Click Install to install the setup support files. (2 minutes)
- 21.** The installer presented us with a warning regarding Windows Firewall. Please consult with your network administrator to ensure that the appropriate ports are open. For now, click Next.
- 22.** On the Feature Selection screen, select only those components you need for this particular installation of SQL Server 2008 Enterprise Edition. This practice reduces overhead on your virtual machine. If you plan to run only the database engine, then select only Database Engine Services | Client Tools Connectivity and Management Tools - Basic.
- 23.** Click Next to continue.
- 24.** On the Instance Configuration screen, either accept the Default instance or specify a named instance. You can have only one default instance on a virtual machine.
- 25.** Click Next to continue.
- 26.** On the Disk Usage Summary screen, click Next.
- 27.** On the Service accounts tab of the Server configuration screen, enter the account name and password you want to use. You must qualify the account name with the domain name: <domain>\<account>. If you want to use a single account for all SQL services, enter the account and password in the Use the same account for all section, and click Apply to all.
- 28.** On the Collation tab of the Server configuration screen, accept the defaults unless you have a specific requirement for other collations, and click Next to continue.
- 29.** On the Account Provisioning tab of the Database Engine Configuration screen, select whether you want to run SQL Server 2008 in mixed mode or Windows Authentication only. We recommend leveraging Active Directory security and selecting Windows Authentication. If, however, you must use SQL Server logins, you must select mixed mode; note that this also includes Windows Authentication functionality. You must specify at least one SQL Server administrator. We clicked Add Current User. If you run SQL

Server 2008 in mixed mode, you must enter the password for the "sa" account here.

- 30.** On the Data Directories tab of the Database Engine Configuration screen, enter the appropriate path names. See the Configuring the drives section for help in selecting physical directory and file locations. If possible, separate the user database directory, user database log directory, and tempdb directory.
- 31.** Click Next to continue.
- 32.** On the Error and Usage Reporting screen, select SQL Server Select Send error reports to Microsoft or your corporate report server, and Send anonymous feature usage data to Microsoft if you want to participate in Microsoft's data collection program.
- 33.** Click Next to continue.
- 34.** On the Installation Rules screen, click OK.
- 35.** The Ready to install screen offers one last chance to review your choices. After you do so, click Install. (8 minutes on the Dell PowerEdge R900; 10 minutes on the Dell PowerEdge R905)
- 36.** On the confirmation screen, click Next.
- 37.** On the Complete screen, note the Summary log file location, and click Close to complete the installation.
- 38.** After the SQL Server 2008 installation process completes, check the Microsoft Downloads Center (www.microsoft.com/downloads) for the latest SQL Server updates, patches, and service packs.
- 39.** If you are running 32bit Windows with AWE enabled for SQL Server, you may find that giving the SQL service account the Lock pages in Memory permission helps performance. To do so, follow these steps:
 - a. Select Start | Administrative Tools | Local Security Policy.
 - b. Expand Local Policies, and click User Rights Assignment.
 - c. Scroll down on the right to Lock Pages in Memory, right-click this entry, and select Properties.
 - d. Click Add User or Group..., and then type `DOMAIN\SQLServiceUser` in the object names area (where DOMAIN is your domain and

SQLServiceUser is the domain user under which SQL Server is running).

- e. Click OK to exit.
- f. Restart the SQL Server service by selecting Start | All Programs | Microsoft SQL Server 2008 | Configuration Tools | SQL Server Configuration Manager. Expand SQL Server Services. Right-click the SQL Server, and select Restart.

- 40.** To enable instant file initialization, you must give the SE_MANAGE_VOLUME_NAME permission to the Windows account under which you run the SQL Server 2008 service:
- a. Click Start.
 - b. Go to Administrative tools | Local Security Policy.
 - c. Click User Rights assignment.
 - d. Scroll down to Perform volume maintenance tasks, and double-click.
 - e. Click Add User or Group, and enter the SQL Server service account name.

Appendix C. Migrating SQL Server 2008 databases from your physical machine to your virtual machine

In this appendix, we give detailed instructions on a basic side-by-side migration of a user database from a physical SQL Server server to a virtual SQL Server server. You can accomplish this task in a variety of ways, including using TSQL commands and automated tools. Here, we discuss implementing the database migration using the graphical interface tools Microsoft provides with SQL Server 2008.

1. Log into Windows on the physical SQL Server server as either an administrative user or a user with full SQL Server rights.
2. Start SQL Server Management Studio by selecting Start | All Programs | Microsoft SQL Server 2008 | SQL Server Management Studio.
3. Enter your server name, and select Windows Authentication.
4. Set the relevant database to single-user mode to ensure no updates occur. In the Object Explorer pane, browse to the Databases node, and right-click the database. Choose Properties.
5. Click Options.
6. Scroll down to Restrict Access and choose SINGLE_USER. Click OK. Accept the warning to disconnect other users.



NOTE: Setting the database to single-user mode immediately disconnects all users, causing them to lose any work in progress. You must notify your users well in advance of setting the database to single-user mode.

7. To back up the database and prepare for migration, in the Object Explorer pane, browse to the Databases folder, and right-click the relevant database, choose Tasks, then choose Back Up...
8. Specify the desired path to your backup file.
9. Modify the optional settings if you wish, such as verifying the backup. This increases backup time but checks the integrity of the backup file. We chose to verify the backup. Optionally, you may

choose to compress the backup file in SQL Server 2008.

10. Click OK to begin the backup. When the backup completes, you will receive a success message. Click OK to close the success message. This completes the process on the physical server.



NOTE: We created the backup in a shared local folder, to simplify moving the database later. Also, make sure that no unwanted destinations remain in the Backup to: box from previous backups. These can cause the backup to split into multiple files, which can cause problems later.

11. Log into Windows on your Hyper-V server.
12. Start Hyper-V manager by clicking Start, then Administrative Tools, then Hyper-V manager.
13. Right-click the virtual machine running SQL Server. If the machine is already running, click Connect. If the machine is not already running, click Connect, then choose Action | Start.
14. Log into the virtual machine with sufficient privileges to restore your database.
15. Start SQL Server Management Studio by selecting Start | All Programs | Microsoft SQL Server 2008 | SQL Server Management Studio.
16. Enter your server name, and select Windows Authentication.
17. Right-click the Databases node in the Object Explorer pane, and choose Restore Database...
18. Enter the database name in the To database field.



NOTE: It is critical to use the same database name as on your physical SQL Server server. If the database name does not match, applications that depend on the database name may break.

19. Click From Device, then click Browse.
20. Click Add to add a backup file.
21. Browse to the appropriate path where your database backup file is located.
22. Select the backup file, and click OK. Click OK once more to return to the Restore Database screen.

- 23.** Click the Restore checkbox beside the backup you just selected.
- 24.** If necessary, click Options, and modify file paths as appropriate for your SQL data and log files on your virtual machine.
- 25.** Click OK to begin the restore. When the restore completes, you will receive a success message. Click OK to clear the success message.
- 26.** After the restore is complete, you must change the database access state back to multi-user by performing the following steps:
 - a. Right-click the database in Object Explorer, and select Properties.
 - b. Click Options on the left side to access database options.
 - c. Scroll down to Restrict Access, and change to MULTI_USER.
 - d. Click OK.

Appendix D. Transferring Windows logins via SQL Server Management Studio

In this appendix, we provide detailed instructions for transferring Windows-based logins from a physical server running SQL Server 2008 to a virtual machine running SQL Server 2008 in an Active Directory environment. For information about transferring logins when using SQL Server authentication, see [Appendix E](#).

1. Log into the server running Hyper-V and start Hyper-V manager by clicking Start | Administrative tools | Hyper-V manager.
2. Right-click the target virtual machine, and choose Connect.
3. Log into the virtual machine with sufficient domain credentials.
4. On the virtual machine, open SQL Server Management Studio by selecting Start | All Programs | Microsoft SQL Server 2008 | SQL Server Management Studio.
5. On the Connect to Server screen, select Database Engine for the server type, enter the computer name of your physical SQL Server 2008 server, and click Connect.
6. Select View | Object Explorer Details to bring up the Object Explorer Details tab.
7. On the Object Explorer pane, expand the Security folder, and then highlight the Logins folder.
8. Add the Login Type column to the display by right-clicking the column-header area in the Object Explorer Details pane and selecting Login Type. Now Windows Logins appear as Login Type 0. Optionally, use the column headers to sort by Default Database, or use the Filter button to find only logins that you are interested in migrating.
9. Using the standard Windows multi-select key combinations (Ctrl key or Shift key), highlight the logins you are interested in migrating.
10. Right-click the highlighted logins, and select Script Login as | CREATE To | New Query Editor Window.

- 11.** Right-click the query window, and choose Connection | Change Connection.
- 12.** On the Connect to Database Engine screen, change the Server name to your virtual SQL Server (i.e., the virtual machine you are currently logged into), and click Connect.
- 13.** Click Execute to run the resulting script.

Appendix E. Transferring SQL Server logins via SQL Server Management Studio

In this appendix, we provide detailed instructions on transferring SQL Server logins from a physical SQL Server 2008 server to a virtual SQL 2008 server. For information about transferring logins when using Windows authentication, see [Appendix D](#).

Transferring logins

1. Log into the server running Hyper-V, and start Hyper-V manager by clicking Start | Administrative tools | Hyper-V manager.
2. Right-click the target virtual machine, and choose Connect.
3. Log into the virtual machine with sufficient domain credentials.
4. On the virtual machine, open SQL Server Management Studio by selecting Start | All Programs | Microsoft SQL Server 2008 | SQL Server Management Studio.
5. On the Connect to Server screen, select Database Engine for the server type, enter the computer name of your physical SQL Server 2008 server, and click Connect.
6. Select View | Object Explorer Details to bring up the Object Explorer Details tab.
7. On the Object Explorer pane, expand the Security folder, and then highlight the Logins folder.
8. Add the Login Type column to the display by right-clicking the column-header area in the Object Explorer Details pane and selecting Login Type. Now SQL Server logins appear as Login Type 2. Optionally, use the column headers to sort by Default Database, or use the Filter button to find only SQL Server logins that you are interested in migrating.
9. Using the standard Windows multi-select key combinations (Ctrl key or Shift key), highlight the SQL Server logins you are interested in migrating.
10. Right-click the highlighted logins, and select Script Login as | CREATE To | New Query Editor Window.

- 11.** Right-click the query window, and choose Connection | Change Connection.
- 12.** On the Connect to Database Engine screen, change the Server name to your virtual SQL Server (i.e., the virtual machine you are currently logged into), and click Connect.
- 13.** Click Execute to run the resulting script.

Mapping transferred logins to database users

After transferring SQL Server logins from physical server to virtual server, you must map those logins to the database users that you migrated during the restore process. This process assumes that you have already migrated the applicable database to the new server.

- 1.** On the virtual machine running SQL Server, log into Windows.
- 2.** Open SQL Server Management Studio by selecting Start | All Programs | Microsoft SQL Server 2008 | SQL Server Management Studio.
- 3.** On the Connect to Server screen, select Database Engine for the server type, enter the computer name of the virtual machine, and click Connect.
- 4.** Right-click the relevant database, and select New Query.
- 5.** Enter the following command, and click Execute:

```
EXEC sp_change_users_login 'Report';
```

- 6.** For each user output from the above step, map these "orphaned" users to the appropriate matching SQL Server login by performing the following steps:
 - a. Open a query window on the SQL Server using steps 1 to 3 above.
 - b. For each login you wish to map, enter the following command, and click Execute:

```
EXEC sp_change_users_login  
'Auto_Fix', 'SQLLogin';
```

- c. The above command maps the database user SQLLogin to the SQL Server login of SQLLogin. This procedure assumes that the database user and SQL Server login have the

same value. Upon successful execution of this procedure, you will see output similar to the following:

The number of orphaned users fixed by updating users was 1.

- d. To reset the password and enable the account, enter the following command, and click Execute:

```
USE [master]
GO
```

```
ALTER LOGIN [SQLLogin] WITH
PASSWORD=N'Password1 '
GO
```

```
ALTER LOGIN [SQLLogin] ENABLE
GO
```

The above commands reset the SQLLogin login to have a password of Password1, and enable that login.

About Principled Technologies



Principled Technologies, Inc.
1007 Slater Road, Suite 250
Durham, NC, 27703
www.principledtechnologies.com

We provide industry-leading technology assessment and fact-based marketing services. We bring to every assignment extensive experience with and expertise in all aspects of technology testing and analysis, from researching new technologies, to developing new methodologies, to testing with existing and new tools.

When the assessment is complete, we know how to present the results to a broad range of target audiences. We provide our clients with the materials they need, from market-focused data to use in their own collateral to custom sales aids, such as test reports, performance assessments, and white papers. Every document reflects the results of our trusted independent analysis.

We provide customized services that focus on our clients' individual requirements. Whether the technology involves hardware, software, Web sites, or services, we offer the experience, expertise, and tools to help you assess how it will fare against its competition, its performance, whether it's ready to go to market, and its quality and reliability.

Our founders, Mark L. Van Name and Bill Catchings, have worked together in technology assessment for over 20 years. As journalists, they published over a thousand articles on a wide array of technology subjects. They created and led the Ziff-Davis Benchmark Operation, which developed such industry-standard benchmarks as Ziff Davis Media's Winstone and WebBench. They founded and led eTesting Labs, and after the acquisition of that company by Lionbridge Technologies were the head and CTO of VeriTest.

Principled Technologies is a registered trademark of Principled Technologies, Inc.
All other product names are the trademarks of their respective owners

Disclaimer of Warranties; Limitation of Liability:

PRINCIPLED TECHNOLOGIES, INC. HAS MADE REASONABLE EFFORTS TO ENSURE THE ACCURACY AND VALIDITY OF ITS TESTING, HOWEVER, PRINCIPLED TECHNOLOGIES, INC. SPECIFICALLY DISCLAIMS ANY WARRANTY, EXPRESSED OR IMPLIED, RELATING TO THE TEST RESULTS AND ANALYSIS, THEIR ACCURACY, COMPLETENESS OR QUALITY, INCLUDING ANY IMPLIED WARRANTY OF FITNESS FOR ANY PARTICULAR PURPOSE. ALL PERSONS OR ENTITIES RELYING ON THE RESULTS OF ANY TESTING DO SO AT THEIR OWN RISK, AND AGREE THAT PRINCIPLED TECHNOLOGIES, INC., ITS EMPLOYEES AND ITS SUBCONTRACTORS SHALL HAVE NO LIABILITY WHATSOEVER FROM ANY CLAIM OF LOSS OR DAMAGE ON ACCOUNT OF ANY ALLEGED ERROR OR DEFECT IN ANY TESTING PROCEDURE OR RESULT.

IN NO EVENT SHALL PRINCIPLED TECHNOLOGIES, INC. BE LIABLE FOR INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH ITS TESTING, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN NO EVENT SHALL PRINCIPLED TECHNOLOGIES, INC.'S LIABILITY, INCLUDING FOR DIRECT DAMAGES, EXCEED THE AMOUNTS PAID IN CONNECTION WITH PRINCIPLED TECHNOLOGIES, INC.'S TESTING. CUSTOMER'S SOLE AND EXCLUSIVE REMEDIES ARE AS SET FORTH HEREIN.
