

CONSOLIDATION



**CONSOLIDATING SQL SERVER® 2005
ONTO DELL™ POWEREDGE™ R900 AND
POWEREDGE™ R905 USING
MICROSOFT'S HYPER-V™**

A Principled Technologies report commissioned by Dell



Table of contents

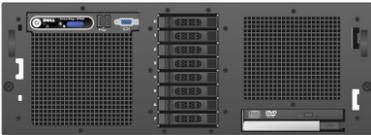
Table of contents	2
Introduction	3
Overview of this companion Guide.....	3
Continuing the migration and consolidation process	4
Creating and configuring your virtual machines	5
Creating a virtual machine	5
Installing Windows Server 2003 in your virtual machine	6
Installing SQL Server 2005 in your virtual machine.....	9
Moving to your virtual environment	13
Moving your database	13
Transferring logins	15
Summing up	16
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 2003 SP2 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 2005 Enterprise Edition.	24
Installing SQL Server 2005 with SP3.....	24
Moving tempdb files from the default location	27
Appendix C. Migrating SQL Server 2005 databases from your physical machine to your virtual machine	29
Appendix D. Transferring Windows logins via SQL Server Management Studio	32
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® 2005 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 both SQL Server 2005 instances onto a single physical system.

This document covers the concepts and procedures specific to virtualizing SQL Server 2005 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 2005. 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 2005 Guide in conjunction with the general Hyper-V overview Guide, you can easily virtualize your SQL Server 2000 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>.

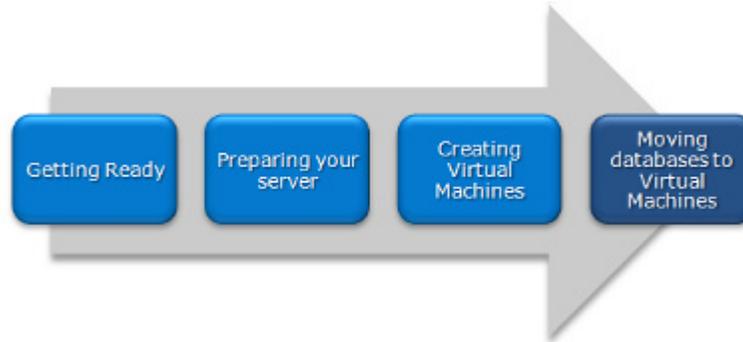
Because we use both the Dell PowerEdge R900 and PowerEdge R905 in this Guide, we refer to the target server as the PowerEdge, and note specifics to each platform where they apply.

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 2005 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.

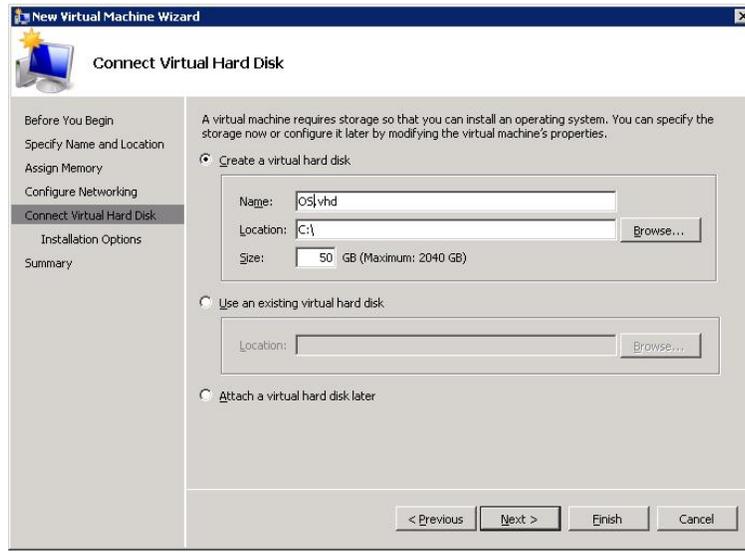


Figure 1. Configuring a Hyper-V virtual machine.

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 2003 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 2003. 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 2003 installation process on a Hyper-V virtual machine. For the most part, this installation on the virtual machine is identical to a typical Windows Server 2003 installation on the physical machine. We include approximate wait times for each step. ([Appendix A](#) provides complete, detailed installation instructions.)



NOTE: Plan on at least 70 minutes for installing Windows Server 2003 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.

1. Insert the Microsoft Windows Server 2003 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 accept the Windows Server license agreement. Create your partition if necessary, format your drive using NTFS, and proceed with the Windows Server 2003 installation steps on your virtual machine (see Figure 2). (12 minutes on the PowerEdge R900; 15 minutes on the PowerEdge R905)

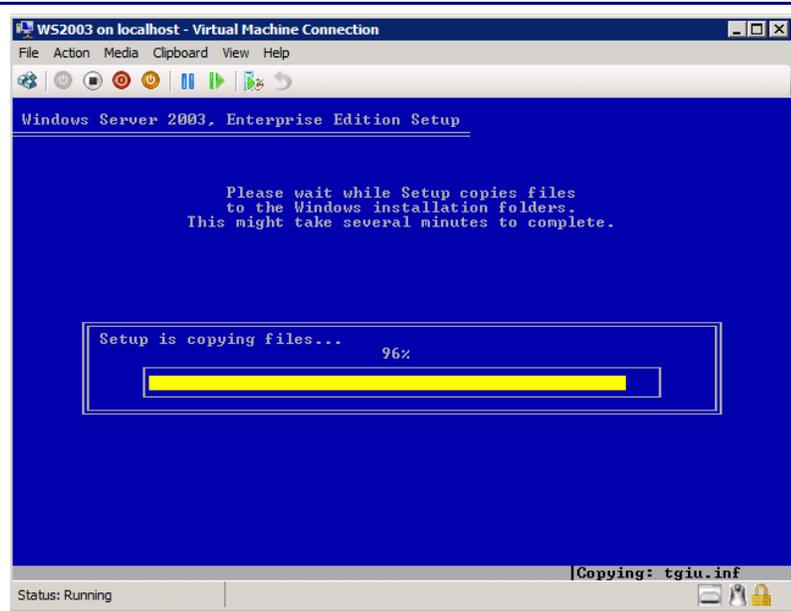


Figure 2. Installing Windows Server 2003 on a virtual machine.

4. Enter your product key and licensing information, set the Administrator password, and set your time and location. Enter any other information the installation software requires. (14 minutes on the PowerEdge R900; 21 minutes on the PowerEdge R905)
5. After the system restarts, complete the installation when Windows Server 2003 prompts you for the second disk. Then, install Windows updates.
6. Enter configuration information, including the IP addresses and domain information, and reboot the system. (3 minutes)
7. Install Service Pack 2. Note: SP2 is a requirement for using the Hyper-V Integration Services. We had previously downloaded Service Pack 2. (8 minutes on the PowerEdge R900; 16 minutes on the PowerEdge R905)
8. 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 the Integration Services installation requires it. (5 minutes)

9. Configure your disks inside the VM operating system. Follow the same procedures you used when configuring your physical disks, except that these are virtual disks, connected to the VHD files you created earlier. (3 minutes per disk, though times vary a bit)

Installing SQL Server 2005 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 2005 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 2005 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 2005 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. Browse to the DVD drive, execute the setup file, accept license agreement info, and install prerequisites.
4. Choose SQL Server components to install. We chose only SQL Server Database Services and Workstation components.
5. Customize security accounts, collation settings, and other details if the setup application requires it.
6. Complete the installation, and reboot if the setup application prompts you to do so.
7. Download SQL Server 2005 SP3 (the latest SP as of this writing), and execute the file to start SP3 installation.
8. SP3 identifies software that it needs to update. Accept the defaults and complete SP3 installation.

9. Reboot if the setup application prompts you to do so.

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 2003 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 3).
5. Click Add User or Group, and enter the SQL Server service account name.

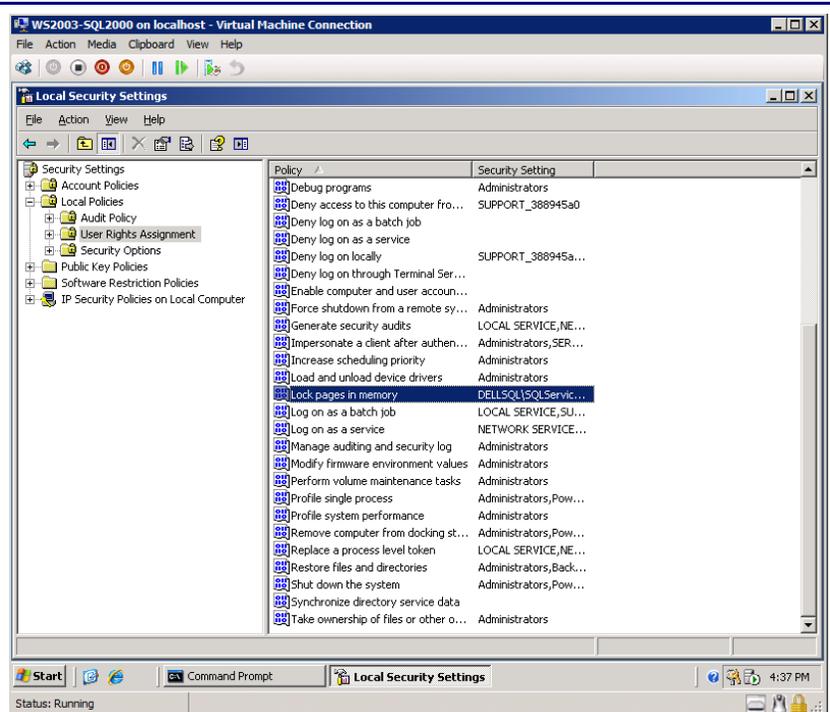


Figure 3. Selecting the Lock pages in memory setting inside your VM.



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.

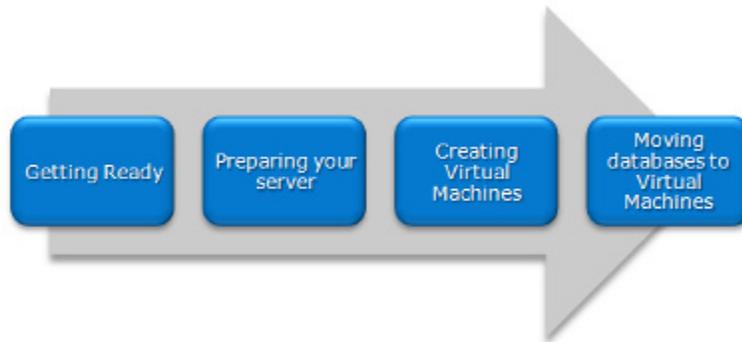
Enable instant file initialization. The default behavior is to initialize the storage with zeros whenever SQL Server creates a data file or allocates an extent to grow a data file. Zero-filling storage can be very time-consuming. When you enable instant file initialization, the system does not initialize 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 2005 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 2005 databases to your virtual machine running SQL Server 2005. 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 2005 backup and restore maintenance using SQL Server Management Studio, the main administration interface for SQL Server 2005.



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 Figure 4). See [Appendix C](#) for details.

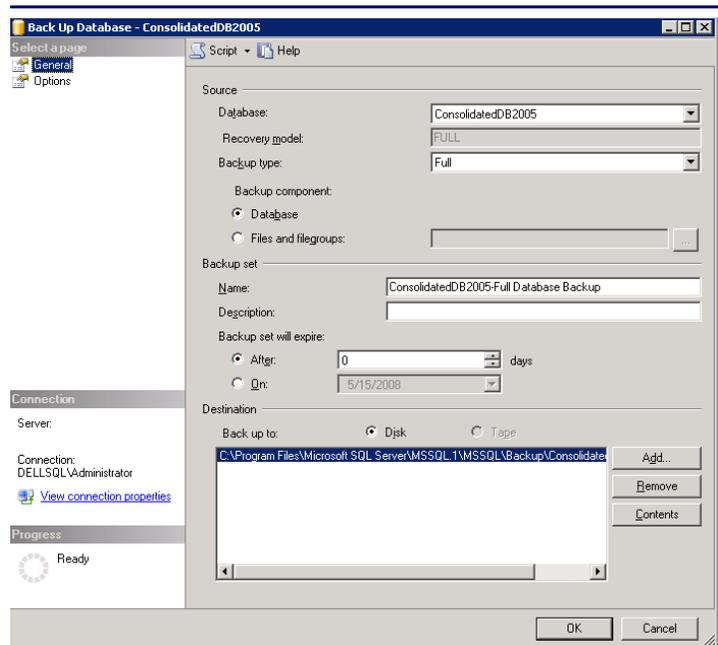


Figure 4. Example SQL Server 2005 backup configuration.



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 [Appendix C](#) for details.



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 system 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 server, do so now by right-clicking and selecting Change Connection.
- 5.** Execute the script on your virtual SQL Server 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 2005 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, but only two logical processors for Windows Server 2003 SP2 32-bit. When we set up the number of processors for Windows Server 2003 SP2, Hyper-V offered four processors as an option. However, as this is not a Microsoft supported configuration, we did not experiment with it, nor do we recommend running with it.

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 2003 SP2 guest OS

Allow 70 minutes to complete the installation of the Windows Server 2003 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 2003 Enterprise Edition CD into the DVD drive. From the Action menu, choose Start. You will see the blue Windows Setup screen as it loads. (2 minutes)
- 4.** On the Welcome to Setup screen, press Enter for a fresh install.
- 5.** Press F8 to accept the license agreement.
- 6.** Press Enter to install Windows on the unpartitioned space.
- 7.** Format the disk as NTFS. You may use the quick format if you choose.
- 8.** After the format, you will boot into the Windows Install. (12 minutes on the PowerEdge R900; 15 minutes on the PowerEdge R905)
- 9.** On the Regional and Languages Options screen, click Next.
- 10.** Enter your user name and organization name, and click Next.
- 11.** Enter your Product key, and click Next.
- 12.** Set the licensing mode as appropriate for your installation, and click Next.
- 13.** Enter a strong password for the administrator, and click Next.
- 14.** Set the date and time, and click Next. The install will continue. (14 minutes on the Dell PowerEdge R900; 21 minutes on the Dell PowerEdge R905)
- 15.** When the installer prompts you, log in using the password you set earlier. If necessary, use the Ctrl+Alt+Del button provided in the virtual machine interface to invoke the login screen.

- 16.** Insert the second Windows Server 2003 CD, and click OK.
- 17.** On the Windows Server 2003 R2 Setup Wizard, click Next.
- 18.** On the setup summary screen, click Next. (2 minutes)
- 19.** When the Setup Wizard is complete, click Finish.
- 20.** Before proceeding, you need to install SP2. We downloaded it and burned it to a CD.
- 21.** Put the SP2 CD in the DVD drive. Double-click the executable to run it. (3 minutes to load)
- 22.** On the SP2 installer splash screen, click Next.
- 23.** Accept the license agreement, and click Next.
- 24.** Click Next to accept the default uninstall folder.
- 25.** SP2 will start updating the VM. (8 minutes on the Dell PowerEdge R900; 16 minutes on the Dell PowerEdge R905)
- 26.** When SP2 finishes installing, restart the VM.
- 27.** Once the VM has rebooted, choose Insert Integration Services Setup Disk from the Action menu. Proceed with Integration Services installation.
- 28.** Turn off the VM. (2 minutes)
- 29.** Now that you have installed the Integration Services, in the Hyper-V manager, right-click the VM, and choose Settings.
- 30.** Click Network adaptor, and select your virtual NIC.
- 31.** Click Apply, and click OK.
- 32.** Right-click the VM, and choose Start.
- 33.** 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.
- 34.** After restarting, configure your virtual NIC, using the steps in the section Setting IP address, subnet mask, and domain information below.

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 | Control Panel | Network Connections | Local Area Connection.

- 2.** Click Properties.
- 3.** In the Local Area Connection Properties dialog, highlight Internet Protocol Version 4 (TCP/IPv4), and click Properties.
- 4.** Click the radio button next to Use the following IP address.
- 5.** Type the IP address you want the server to have, the Subnet mask, and the Preferred DNS server, and click OK.
- 6.** Close the Network Connections window.
- 7.** From the Start menu, right-click My Computer, and select Properties.
- 8.** Select the Computer Name tab.
- 9.** Click Change.
- 10.** Enter the server name.
- 11.** Click the Domain radio button.
- 12.** Enter the domain name.
- 13.** Enter the user name and password when the system prompts you to do so. (less than 1 minute)
- 14.** When the Computer Name/Domain Changes window appears, click OK to close it.
- 15.** Click OK to go past the warning to restart.
- 16.** Click OK to close the System Properties window.
- 17.** 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 hard drives on the server 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 2005 Enterprise Edition

Install Microsoft SQL Server 2005 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. As of this writing, the latest SP for SQL Server 2005 was SP3.



NOTE: We installed SQL Server 2005 Enterprise Edition using an MSDN distribution, and our steps below reflect that specific distribution.

Installing SQL Server 2005 with SP3

1. Start Hyper-V manager by clicking Start | Administrative Tools | Hyper-V manager.
2. In the Hyper-V manager, right-click the Windows Server 2003 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. Insert the SQL Server 2005 DVD into the DVD drive.
5. Choose Action, then Start. The Hyper-V splash screen appears, followed by the Windows Server 2003 splash screen.
6. Click the Ctrl+Alt+Del button in the virtual machine menu to log onto the VM. Provide the proper administrative credentials to log on.
7. If your CD drive is not set to autoplay, then in Windows Explorer, browse to the CD drive and double-click Default.htm for the top-level SQL Server 2005 install menu. Accept the warnings regarding the display of Active Content.
8. Click SQL Server 2005 Enterprise Edition. Note that if you were installing on a guest OS that was running 64-bit software, then you would choose the 64-bit edition of SQL Server. In our case, we chose the 32-bit edition.
9. Double-click the servers folder, then, double-click setup.exe to start the installation.
10. Accept the licensing terms, and click Next.
11. Click Install to install prerequisites. (3 minutes)

- 12.** Click Next to continue after prerequisite installation completes.
- 13.** On the Welcome to the Microsoft SQL Server Installation Wizard screen, click Next.
- 14.** On the System Configuration Check screen, ensure your server passes all prerequisites. The Internet Information Services (IIS) warning is applicable only if you plan to run SQL Server Reporting Services. Click Next.
- 15.** On the Registration Information screen, enter your name, company name, and product key, if applicable. We did not need to enter the product key, as we had a MSDN distribution. Click Next.
- 16.** On the Components to install screen, for a basic install, select SQL Server Database Services and Workstation Components. Click Next.
- 17.** On the Instance Name screen, keep Default Instance checked, which is the default action. Click Next.
- 18.** On the Service Account Screen, configure the security accounts for SQL Server to use. If installing in a Windows domain, then use a domain user account. If not, use the built-in System account. Click Next.
- 19.** On the Authentication Mode screen, choose either Windows Authentication or mixed mode. If you have SQL Server-based logins, this requires mixed mode. Mixed-mode requires a password for the "sa" account. We chose mixed mode. Click Next.
- 20.** On the Collation Settings screen, keep the default collation, unless otherwise required. Click Next.
- 21.** On the Error and Usage Report Settings screen, click Next.
- 22.** Click Install to begin the installation (10 minutes).
- 23.** After installation completes, click Next, then Finish to exit the installation routine. Reboot the VM.
- 24.** Insert the SQL Server 2005 SP3 CD, or download the file from Microsoft downloads. We used the file download and copied the file to the VM. Double-click the exe file to launch the service pack installation. (3 minutes)
- 25.** On the SP3 Welcome screen, click Next.

- 26.** Accept the license agreement, and click Next.
- 27.** The setup routine finds applicable SQL Server 2005 software to update. Click Next.
- 28.** On the Authentication screen, keep the defaults, and click Next.
- 29.** On the Error and Usage Reporting Settings screen, click Next.
- 30.** On the Running Processes screen, click Next.
- 31.** On the Ready to Install screen, click Install. (8 minutes)
- 32.** Click Next to complete the SP3 installation.
- 33.** Unless you require it, uncheck the Launch the User Provisioning Tool option, and click Finish.
- 34.** If you are running 32-bit Windows with AWE enabled, you may find that giving the SQL service account the Lock pages in memory permission helps performance. To do so, perform the following 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 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 2005 | Configuration Tools | SQL Server Configuration Manager. Expand SQL Server Services. Right-click the SQL Server service, and select Restart.
- 35.** 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 2005 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.

Moving tempdb files from the default location

The SQL Server 2005 installation routine stores tempdb by default on the on the C: drive, or wherever you specified during the installation routine with other user data. Our best practices recommend that you place tempdb on separate physical devices from user data for performance, unless your situation does not allow for this. Here, we show how to move the tempdb database and its associated files to another volume.

1. In your virtual machine, click Start | All Programs | Microsoft SQL Server 2005 | SQL Server Management Studio.
2. Enter your virtual machine name, choose Windows Authentication, and click Connect.
3. Click the New Query button.
4. Type in the following command to retrieve the metadata related to tempdb. Press F5 to execute the command.

```
sp_helpdb tempdb
GO
```

5. Note the name column and the filename column.
6. Delete the previous command from the query window, and use the following commands to reconfigure the tempdb file locations. Note that the paths listed below are simply placeholders, and that you should modify them to fit your situation. Press F5 to execute these commands.

```
ALTER DATABASE tempdb MODIFY FILE ( NAME
= tempdev,
FILENAME = 'E:\newtempdbpath\tempdb.mdf'
)
GO
ALTER DATABASE tempdb MODIFY FILE ( NAME
= templog, FILENAME =
'E:\newtempdbpath\templog.ldf' )
GO
```

- 7.** Restart the SQL Server service to make these changes permanent. Click Start | All Programs | Microsoft SQL Server 2005 | Configuration Tools | SQL Server Configuration Manager
- 8.** Click SQL Server 2005 Services.
- 9.** Right-click the SQL Server service and choose restart. Wait for the service to stop and start.
- 10.** Remove the original tempdb files, as the SQL Server instance has now recreated the tempdb files in the newly specified location.

Appendix C. Migrating SQL Server 2005 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 2005.

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 2005 | 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.
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 2005 | 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 2005 to a virtual machine running SQL Server 2005 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 2005 | 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 2005 server, and click Connect.
- 6.** Select View | Object Explorer Details to bring up the Object Explorer Details tab.
- 7.** In the Object Explorer pane, expand the Security folder, and then highlight the Logins folder.
- 8.** Optionally, use the column headers to sort by Name, 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 Windows 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 2005 server to a virtual SQL 2005 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 2005 | 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 2005 server, and click Connect.
6. Select View | Object Explorer Details to bring up the Object Explorer Details tab.
7. In the Object Explorer pane, expand the Security folder, and then highlight the Logins folder.
8. 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 2005 | 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.
