Realtime
publishers

Automating Windows 7 Installation for Desktop and VDI Environments

Greg Shields

Copyright Statement

Chapter 6: Automating Application Inventory and Overcoming Incompatibility

Step Eleven, covered in the previous chapter, really concludes the "deployment" activities in Windows 7 deployment. By navigating through the first eleven steps, you know how to create and deploy images. You've seen how to layer applications on top of those images during deployment. You've also learned tricks you'll need to preserve important user data between old and new OS instances.

Congratulations! You're ready to start deploying!

Or are you? If you've been around the IT industry for long, you know that every OS upgrade is never as simple as it seems. Although deploying OSs is pretty easy, making sure applications work with that OS often isn't. Here's a chilling statement: *Many of your applications that run just fine on Windows XP won't automatically run on Windows 7.* Yikes.

You should already know some of the reasons that this is the case. Windows Vista and Windows 7 arrive with significant changes to their security model. These two OSs no longer grant applications and drivers direct access to the OS's most-secure kernel code. The new security model along with a host of other updates make Windows Vista and Windows 7 significantly more secure than Windows XP. *Significantly.*

Unfortunately, those changes also mean many older applications—those that abused Windows XP's overly-open and inviting nature—aren't going to work automatically. Some will require configuration adjustments. Others will need software "shims." Even others won't work at all.

Thankfully, Microsoft knows about the problem. To assist, they've created two tools (along with accompanying acronyms) that inventory your network and apply known fixes that'll make many applications work again. The first of these tools is called the *Microsoft Assessment and Planning Toolkit* (called the MAP). A simple tool, the MAP collects a list of applications and drivers on your network. With its data, you can identify the applications on your network and a list of drivers your computers will need. The second and more powerful tool is the *Microsoft Application Compatibility Toolkit* (aka the ACT), which assists in identifying and fixing incompatible applications.

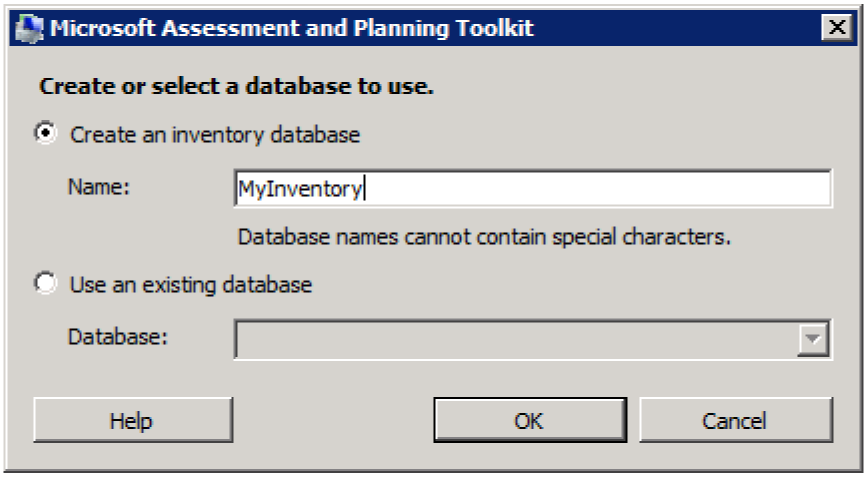
Step Twelve: Inventorying Applications and Drivers on the Network

Step Twelve begins with the installation of the MAP. With it, I'll show you how to gather a report of the software that's installed on computers around your network. With that information in hand, I'll point you to Microsoft's Windows 7 Compatibility Center. This site is an online clearinghouse of applications and their compatibility status. You can compare the software in your report to those in the Compatibility Center to see which will work and which won't.

But that's not all the MAP is good for. Remember back in Step Four (and again in Step Eight) where I showed you how drivers can be automatically injected into images as they're deployed? Wouldn't it be useful to know exactly which drivers your computers will need? The MAP can also collect that information for you, if you know where to look.

Installing the MAP and Collecting Inventory

Begin by downloading the MAP from Microsoft's Web site and installing it to the WDS server we've been using throughout this book. Using the MAP requires first installing a copy of Microsoft Office 2007 SP2 as well as the .NET Framework. The MAP will automatically install a copy of SQL Server Express to the computer as it begins its installation. Once installed, you'll be asked to create a new inventory database just like you see in Figure 6.1. I've named my database *MyInventory*.


Microsoft Assessment and Planning Toolkit

Create or select a database to use.

● Create an inventory database

Name: MyInventory

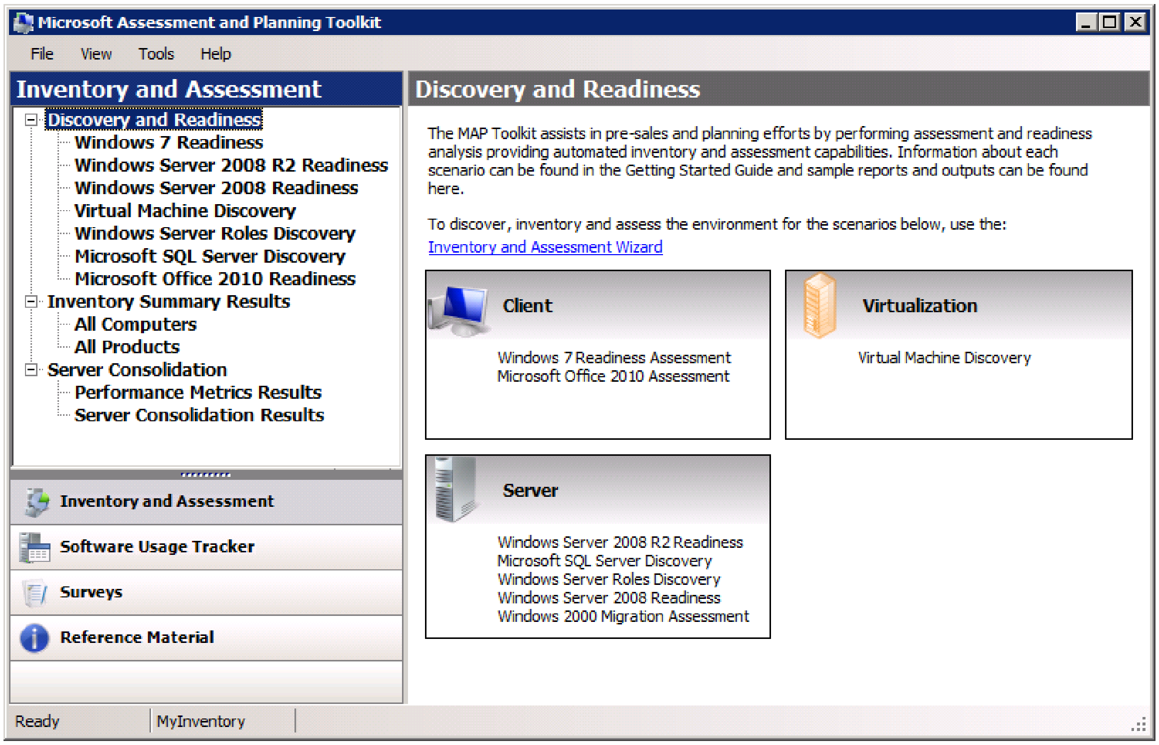
Database names cannot contain special characters.

○ Use an existing database

Database:

Figure 6.1: Creating a new MAP database.

Figure 6.2 shows what the MAP's console will look like after installation. You should immediately notice that the MAP has far more capabilities than simply searching your network for installed software. Other assessments are available that help determine Windows Server roles that have been installed on servers, where SQL server components have been deployed, and even where virtual machines might be hiding on your network.


Microsoft Assessment and Planning Toolkit
File
View
Tools
Help
Inventory and Assessment
Discovery and Readiness
Windows 7 Readiness
Windows Server 2008 R2 Readiness
Windows Server 2008 Readiness
Virtual Machine Discovery
Windows Server Roles Discovery
Microsoft SQL Server Discovery
Microsoft Office 2010 Readiness
Inventory Summary Results
All Computers
All Products
Server Consolidation
Performance Metrics Results
Server Consolidation Results
Inventory and Assessment
Software Usage Tracker
Surveys
Reference Material
Ready
MyInventory
Discovery and Readiness

The MAP Toolkit assists in pre-sales and planning efforts by performing assessment and readiness analysis providing automated inventory and assessment capabilities. Information about each scenario can be found in the Getting Started Guide and sample reports and outputs can be found here.

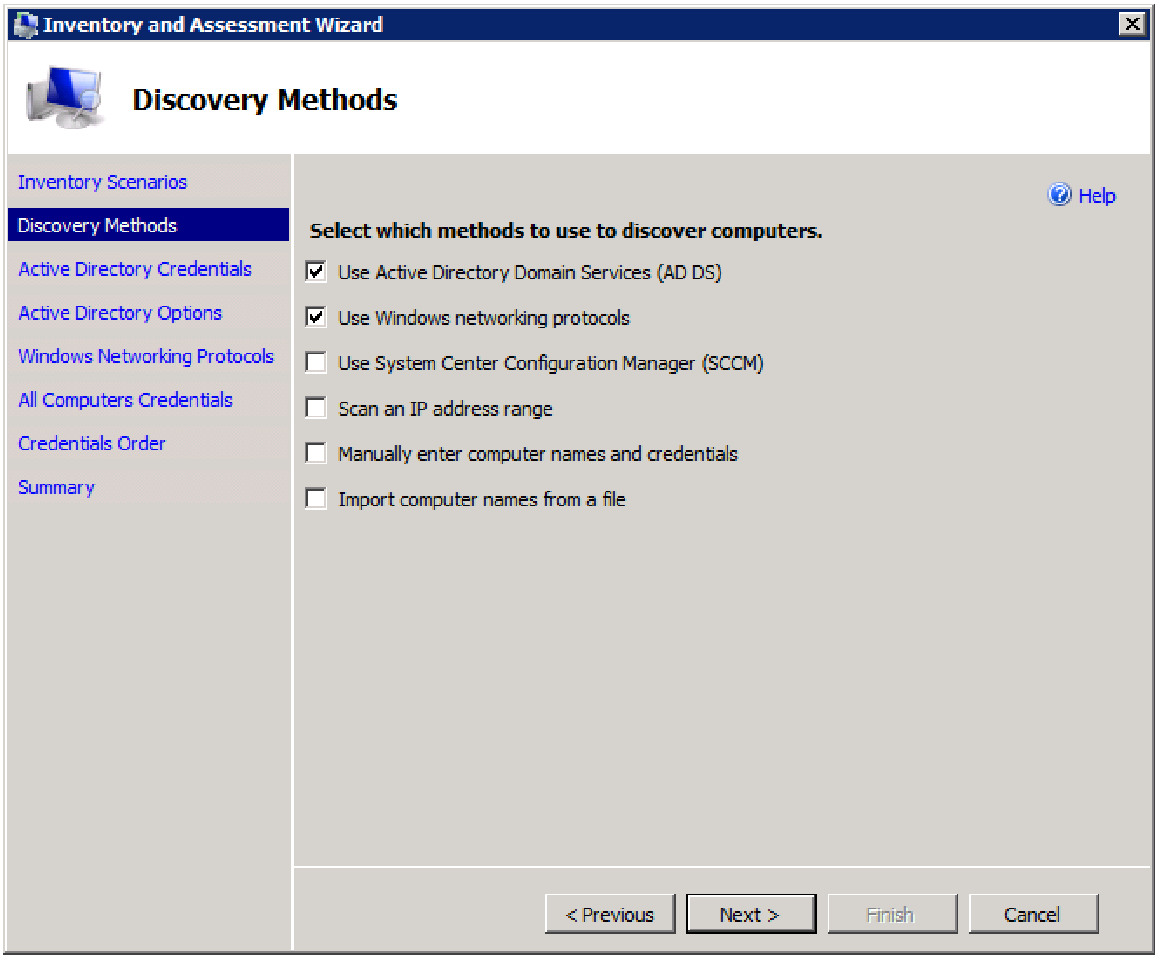
To discover, inventory and assess the environment for the scenarios below, use the:
Inventory and Assessment Wizard

Client
Windows 7 Readiness Assessment
Microsoft Office 2010 Assessment
Virtualization
Virtual Machine Discovery
Server
Windows Server 2008 R2 Readiness
Microsoft SQL Server Discovery
Windows Server Roles Discovery
Windows Server 2008 Readiness
Windows 2000 Migration Assessment

Figure 6.2: The MAP console.

Inventorying the software in your environment starts by clicking the Inventory and Assessment Wizard link you see in the right pane of Figure 6.2. Clicking this link brings forward a wizard that you'll use to configure the types of inventory to be collected. Windows, Linux, VMware, Exchange Server, and SQL Server computers are all options for inventorying. I'll be using only the Windows-based computers scenario in this chapter, as this scenario provides the information I'll need for a Windows 7 upgrade.

The wizard's second page (seen in Figure 6.3) shows the multitude of methods the MAP will use in discovering computers to inventory. My computers are all members of an Active Directory (AD) domain, so I can select the first and second check boxes to find them. Other computers not on the domain can be discovered via IP ranges, by entering in computer names manually, or via a text file.



Select which methods to use to discover computers.

☑ Use Active Directory Domain Services (AD DS)

☑ Use Windows networking protocols

☐ Use System Center Configuration Manager (SCCM)

☐ Scan an IP address range

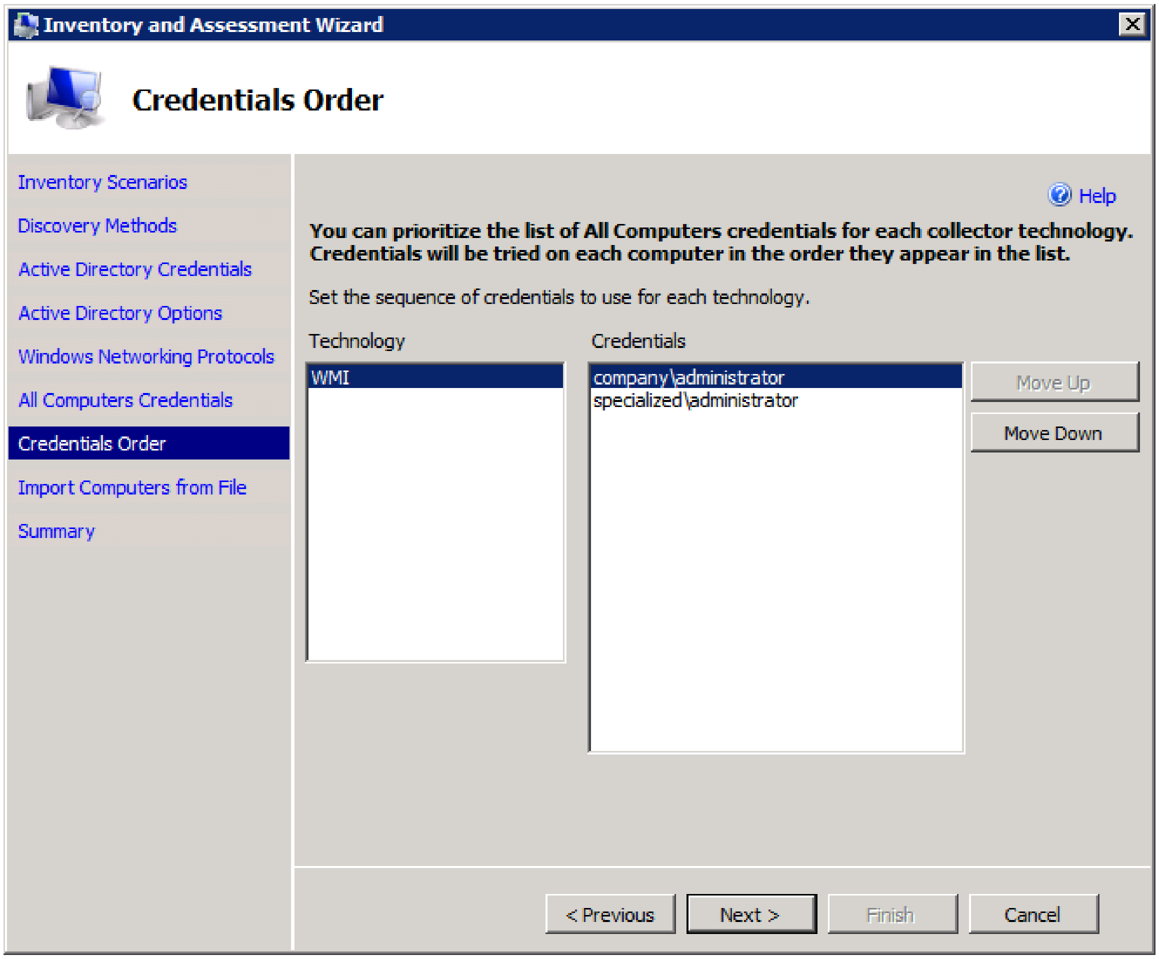
☐ Manually enter computer names and credentials

☐ Import computer names from a file

Figure 6.3: Selecting methods to discover computers.

Subsequent wizard pages provide locations to enter AD credentials, to restrict inventory to specific organizational units (OUs), and to add domains or workgroups if they are discovered by the tool. The page titled All Computers Credentials allows you to enter in a list of possible credentials the tool can use in attempting to inventory discovered computers.

It is within the All Computers Credentials and Credentials Order pages where the MAP truly shines. You can see in Figure 6.4 that I have entered credentials for two different domains: COMPANY and SPECIALIZED. Additional workgroups or specific computer credentials can be added as well. Doing so will give the inventory process plenty of username and password options as it authenticates to discovered computers.


Inventory and Assessment Wizard
Inventory Scenarios
Discovery Methods
Active Directory Credentials
Active Directory Options
Windows Networking Protocols
All Computers Credentials
Credentials Order
Import Computers from File
Summary
Help
Move Up
Move Down
< Previous
Next >
Finish
Cancel

Credentials Order

You can prioritize the list of All Computers credentials for each collector technology. Credentials will be tried on each computer in the order they appear in the list.

Set the sequence of credentials to use for each technology.

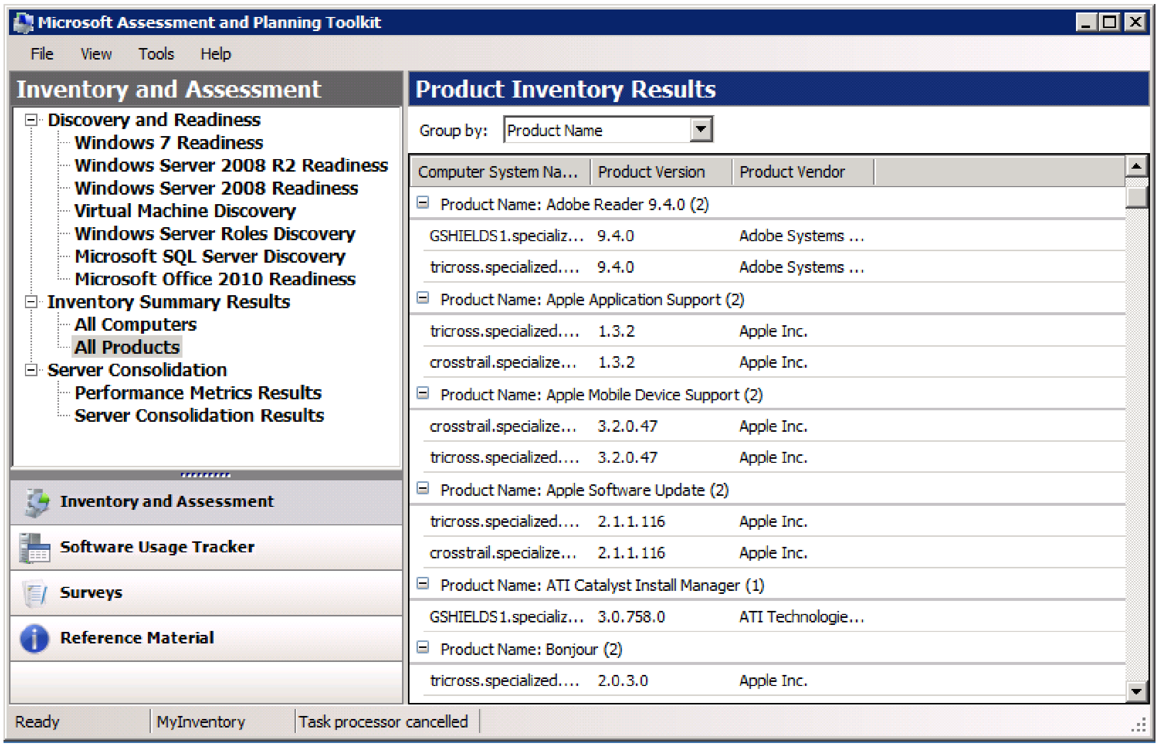
Technology	Credentials
WMI	company\administrator specialized\administrator

Figure 6.4: Setting an order of credentials.

Click Finish to complete the wizard and begin the discovery and inventory process. Be aware that this process can take a considerable quantity of time, particularly if your scope is large. Version 5.0 of the MAP, the version used in this example, is reported to discover and inventory as many as 100,000 computers. Gathering information from that quantity of computers, as you can imagine, is going to take a while.

> **Note**
> The MAP's inventory process uses WMI queries to gather its information. Ensure that the Remote Administration firewall exception has been enabled on any computers that will be queried by the MAP.

Figure 6.5 shows a report on the products the MAP found in my network. You can see that the Adobe Reader 9.4.0 was discovered on two computers. A set of three Apple applications was found on another two, as well as an entire list of software from all sorts of vendors. This screen inside the report is relatively static, giving you little more than a view of the software that the MAP has found inside your network.


Microsoft Assessment and Planning Toolkit
File
View
Tools
Help
Inventory and Assessment
Discovery and Readiness
Windows 7 Readiness
Windows Server 2008 R2 Readiness
Windows Server 2008 Readiness
Virtual Machine Discovery
Windows Server Roles Discovery
Microsoft SQL Server Discovery
Microsoft Office 2010 Readiness
Inventory Summary Results
All Computers
All Products
Server Consolidation
Performance Metrics Results
Server Consolidation Results
Inventory and Assessment
Software Usage Tracker
Surveys
Reference Material
Product Inventory Results

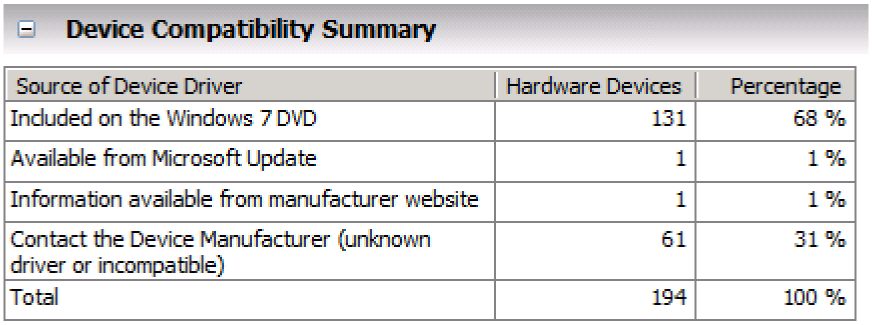
Group by: Product Name

Computer System Na...	Product Version	Product Vendor
Product Name: Adobe Reader 9.4.0 (2)		
GSHIELDS1.specializ...	9.4.0	Adobe Systems ...
tricross.specialized....	9.4.0	Adobe Systems ...
Product Name: Apple Application Support (2)		
tricross.specialized....	1.3.2	Apple Inc.
crosstrail.specialize...	1.3.2	Apple Inc.
Product Name: Apple Mobile Device Support (2)		
crosstrail.specialize...	3.2.0.47	Apple Inc.
tricross.specialized....	3.2.0.47	Apple Inc.
Product Name: Apple Software Update (2)		
tricross.specialized....	2.1.1.116	Apple Inc.
crosstrail.specialize...	2.1.1.116	Apple Inc.
Product Name: ATI Catalyst Install Manager (1)		
GSHIELDS1.specializ...	3.0.758.0	ATI Technologie...
Product Name: Bonjour (2)		
tricross.specialized....	2.0.3.0	Apple Inc.

Ready | MyInventory | Task processor cancelled

Figure 6.5: Product inventory results.

A much more useful representation of the data found by the MAP can be created by clicking the *Windows 7 Readiness* link in Figure 6.5's left pane. The resulting Windows 7 Readiness summary provides high-level information about the computers found in the discovery and inventory process. You can learn in this screen how many computers have hardware that is powerful enough to support Windows 7. You can also learn how many drivers your computers will need and which of those drivers are included on the Windows 7 DVD. Figure 6.6 shows a snippet of the summary screen. That screen tells me I'll need to locate manufacturer drivers for 61 of the 194 drivers my computers say they need.



Device Compatibility Summary

Source of Device Driver	Hardware Devices	Percentage
Included on the Windows 7 DVD	131	68 %
Available from Microsoft Update	1	1 %
Information available from manufacturer website	1	1 %
Contact the Device Manufacturer (unknown driver or incompatible)	61	31 %
Total	194	100 %

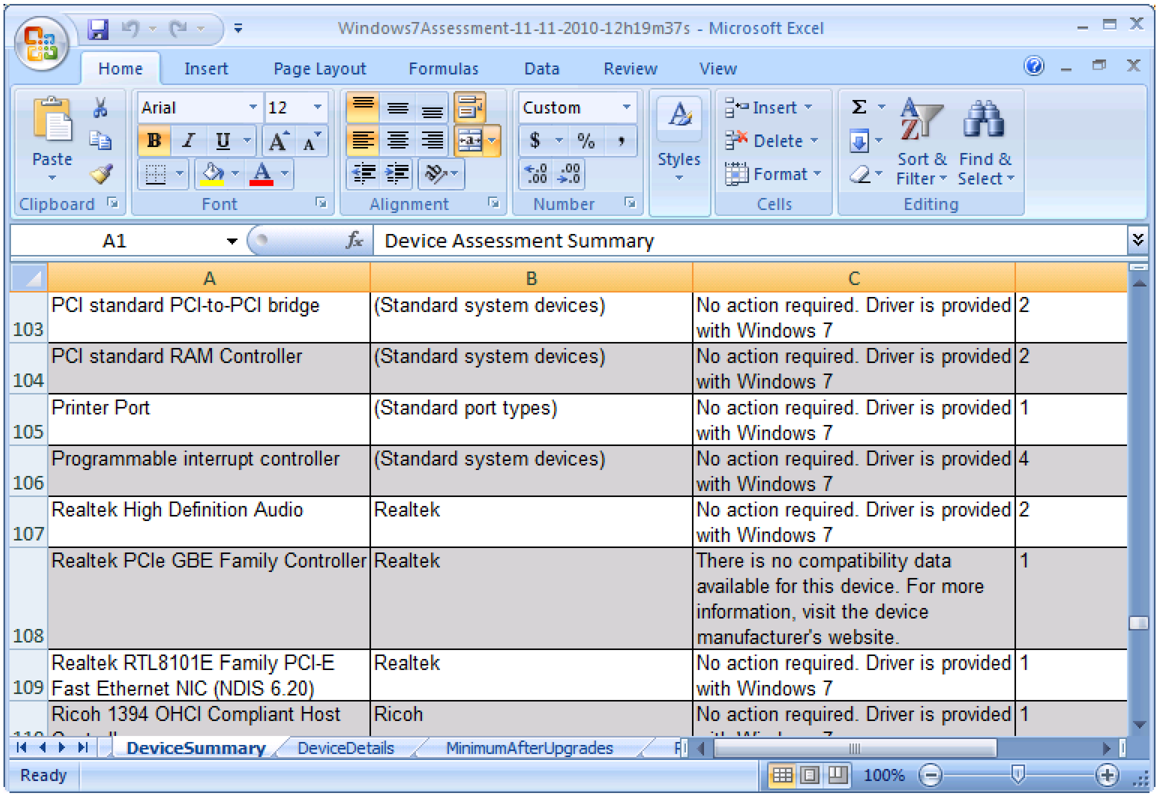
Figure 6.6: Device compatibility summary.

Creating and Using MAP Reports

There's a link (not shown) on the right pane of this Windows 7 Readiness Summary Results page that's labeled Generate Report/Proposal. Click that link to generate a report. Then click View | Saved Reports and Proposals to bring forward an Explorer window. In this window, you'll find a Microsoft Word document that contains useful project planning information about your Windows 7 readiness.

You'll find even more useful information in the accompanying Excel spreadsheet. Inside that spreadsheet is detailed information about each inventoried computer, its hardware configuration, and any installed software and drivers.

Figure 6.7 shows one of the tabs in that spreadsheet. In it you can see that at least one computer on my network reports it will need the *Realtek High Definition Audio* driver. Happily, that driver is available on the Windows 7 DVD media, so I don't need to worry about it. Another computer reports it needs the *Realtek PCIe GBE Family Controller*, which isn't on the Windows 7 media. I'll need to locate that driver from its manufacturer's Web site and add it to my Out-of-Box Drivers node in my MDT Deployment Share.



A1 Device Assessment Summary

	A	B	C	
103	PCI standard PCI-to-PCI bridge	(Standard system devices)	No action required. Driver is provided with Windows 7	2
104	PCI standard RAM Controller	(Standard system devices)	No action required. Driver is provided with Windows 7	2
105	Printer Port	(Standard port types)	No action required. Driver is provided with Windows 7	1
106	Programmable interrupt controller	(Standard system devices)	No action required. Driver is provided with Windows 7	4
107	Realtek High Definition Audio	Realtek	No action required. Driver is provided with Windows 7	2
108	Realtek PCIe GBE Family Controller	Realtek	There is no compatibility data available for this device. For more information, visit the device manufacturer's website.	1
109	Realtek RTL8101E Family PCI-E Fast Ethernet NIC (NDIS 6.20)	Realtek	No action required. Driver is provided with Windows 7	1
110	Ricoh 1394 OHCI Compliant Host	Ricoh	No action required. Driver is provided	1

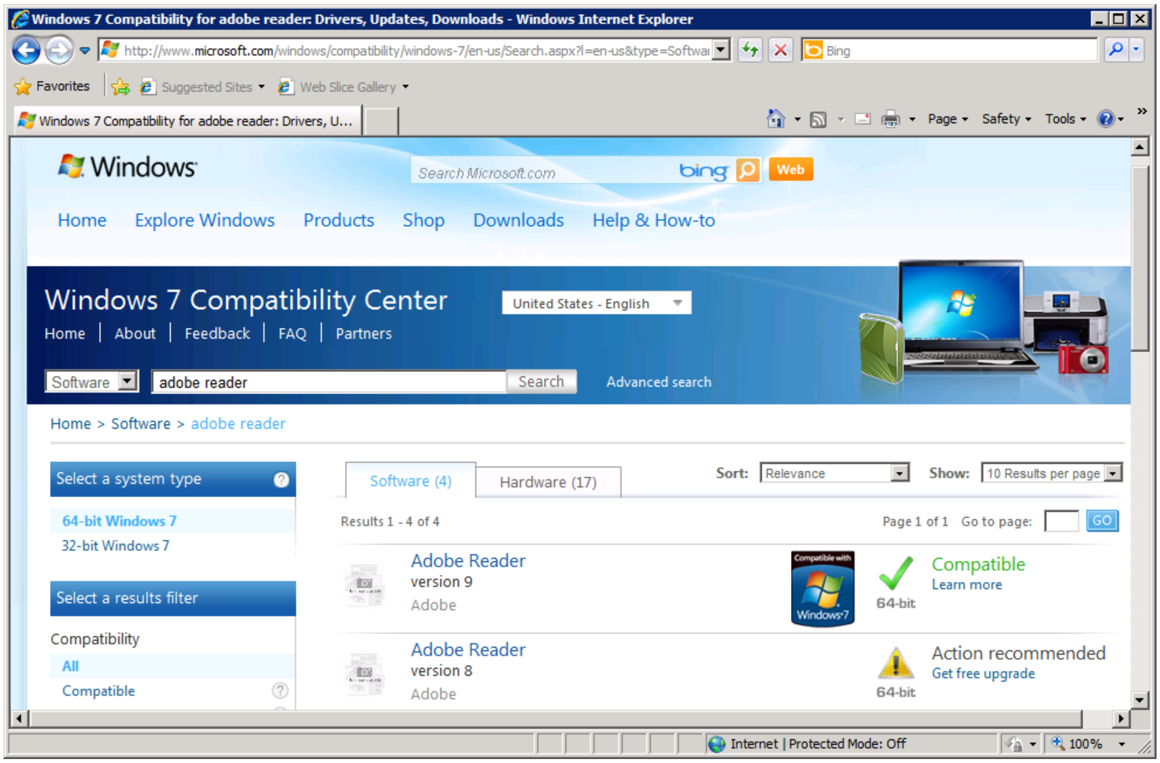
Figure 6.7: A MAP report's Excel spreadsheet.

By reviewing the drivers inside this Excel spreadsheet, I now know which drivers I'll need to make available in MDT so that my images will deploy correctly. This report all by itself gives me the data I need to ensure my deployment goes as smoothly as possible.

A second tab on this Excel spreadsheet gives me a punch list for tracking down the compatibility status of applications that are installed on my computers. That tab, labeled Discovered Applications, lists each application, its version number, and the number of instances found on the network during the last inventory pass.

I mentioned at the beginning of this step that Microsoft has created an online clearinghouse of application compatibility status information. That clearinghouse is called the Windows 7 Compatibility Center. Navigate to http://www.microsoft.com/windows/compatibility/windows-7/en-us/default.aspx to check out its constantly-updated list.

Figure 6.8 shows the results of going to the site and running a search on Adobe reader. I already know from my MAP report that I have two copies of Adobe Acrobat 9.4.0 in my network. Running this search tells me that Adobe Reader version 9 is compatible with Windows 7. It also tells me that version 8 compatibility requires an action, specifically a free upgrade. That's useful information.



Windows 7 Compatibility for adobe reader: Drivers, Updates, Downloads - Windows Internet Explorer

http://www.microsoft.com/windows/compatibility/windows-7/en-us/Search.aspx?l=en-us&type=Softwa

Software | adobe reader | Search

Home > Software > adobe reader

Results 1 - 4 of 4

Adobe Reader
version 9
Adobe
Compatible
Learn more
64-bit

Adobe Reader
version 8
Adobe
Action recommended
Get free upgrade
64-bit

Figure 6.8: The Windows 7 Compatibility Center.

Combining the information provided by this Web site with the information in my MAP report, I can quickly identify which applications will work and which won't. For some, I may learn that they'll require a patch or some other special configuration to function.

Step Thirteen: Resolving Application Incompatibilities

I told you that two tools are necessary to fill out Windows 7's application compatibility story. The second of these two tools is the *Application Compatibility Toolkit (ACT)*. While the MAP is a simple tool that gathers a relatively simple set of information, the ACT is a robust solution for locating and applying fixes to applications that otherwise won't work.

That said, using the ACT isn't nearly as simple as the MAP. You can't just install it and immediately run an inventory. Rather than using WMI to query computers for their contents, the ACT gathers its information through the use of an agent. That agent allows the ACT to gather a greater level of detail than MAP can get with its WMI approach. Although great for data collection, this agent isn't exactly trivial to install. It must be deployed and launched through an external software delivery mechanism. That mechanism can be a logon script, Group Policy Software Installation, or even System Center Configuration Manager. You can even double-click and invoke the agent manually on any computers you want to inventory. Launching that agent through any of these means will gather the data on the system and transfer it to a file share that you designate on your network. For Step Thirteen, I'll show you how to set up the ACT, gather information from agents, and use that information to locate fixes for known-incompatible applications.

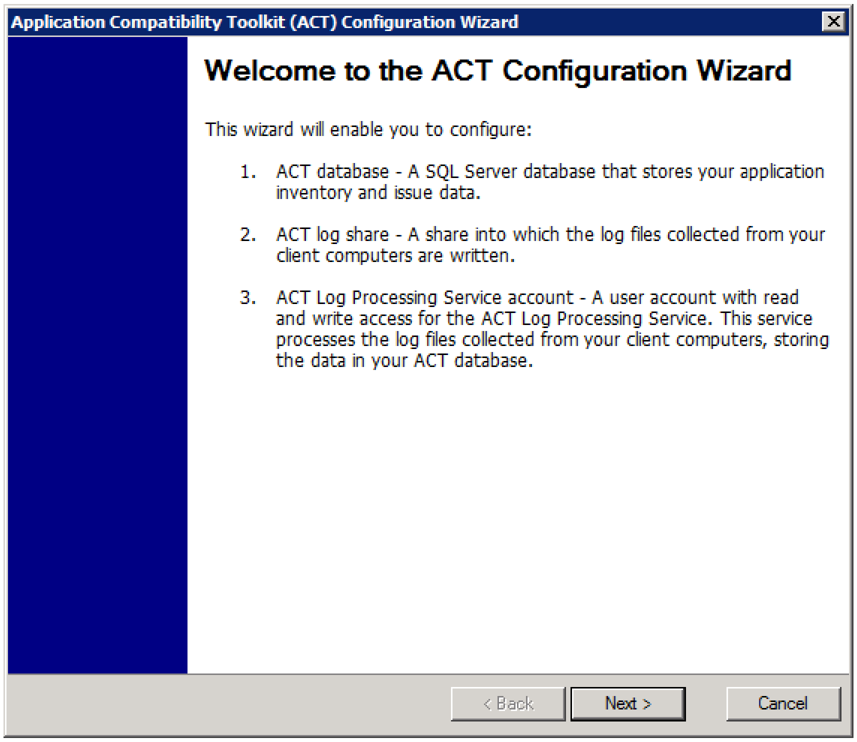
> **Note**
>
> In the interest of space, I'll assume that you know how the steps necessary to automatically deploy a piece of software through one of the aforementioned tools. Should you need assistance, consult Microsoft's deployment guide with the ACT that walks you through the process.

Installing the ACT

Start by downloading the MAP from Microsoft's Web site. In my case, I'll be using the MAP version 5.6. Install it to the server we've been using throughout this book, and launch its Application Compatibility Manager to begin the initial configuration.

> **Note**
>
> If you're using the ACT on the same server where you installed the MAP, you'll need to perform one step prior to launching the Application Compatibility Manager. Navigate to C:\SQLEXPRESS and double-click the installation file you find. Doing so will launch the SQL Server setup wizard. You'll need to install an additional SQL Server instance for the ACT to use. Click through the wizard screens and create a new named instance. In my case, I'll name that instance *ACT*. You'll use that instance in the following steps.

Three information pieces are required to initially configure the ACT (see Figure 6.9). You'll need a SQL Server database instance to store its information. You'll also need a file share with plenty of storage space, which will be used for collecting agent log files. Lastly, you'll need an account that is used for the ACT Log Processing Service.


Application Compatibility Toolkit (ACT) Configuration Wizard

Welcome to the ACT Configuration Wizard

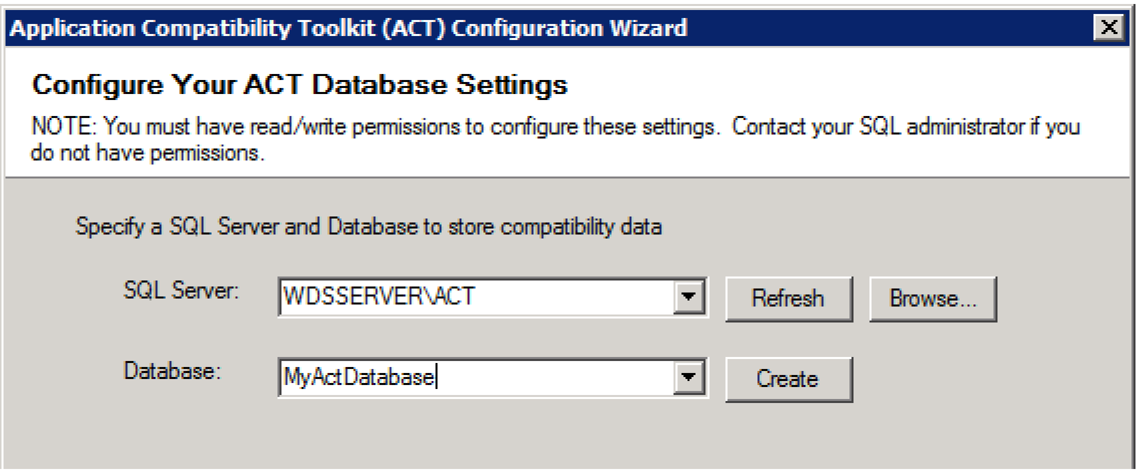
This wizard will enable you to configure:

1. ACT database - A SQL Server database that stores your application inventory and issue data.
2. ACT log share - A share into which the log files collected from your client computers are written.
3. ACT Log Processing Service account - A user account with read and write access for the ACT Log Processing Service. This service processes the log files collected from your client computers, storing the data in your ACT database.

Figure 6.9: The ACT Configuration Wizard.

The ACT Configuration Wizard will step you through setting up these three pieces. This will be your primary server for collecting and processing log files as well as viewing reports, so you'll want to use the ACT's Enterprise configuration when prompted.

Configuring the ACT database requires a somewhat confusing wizard page (see Figure 6.10). Clicking the drop-down box should provide a list of local SQL Server Express instances (see the earlier note for an extra step if you're using the same server where the MAP is installed). Select the instance you want, and click Connect (not shown) to connect. Then enter a database name, and click Create to create that database. I've named my database *MyActDatabase*.


Application Compatibility Toolkit (ACT) Configuration Wizard

Configure Your ACT Database Settings

NOTE: You must have read/write permissions to configure these settings. Contact your SQL administrator if you do not have permissions.

Specify a SQL Server and Database to store compatibility data

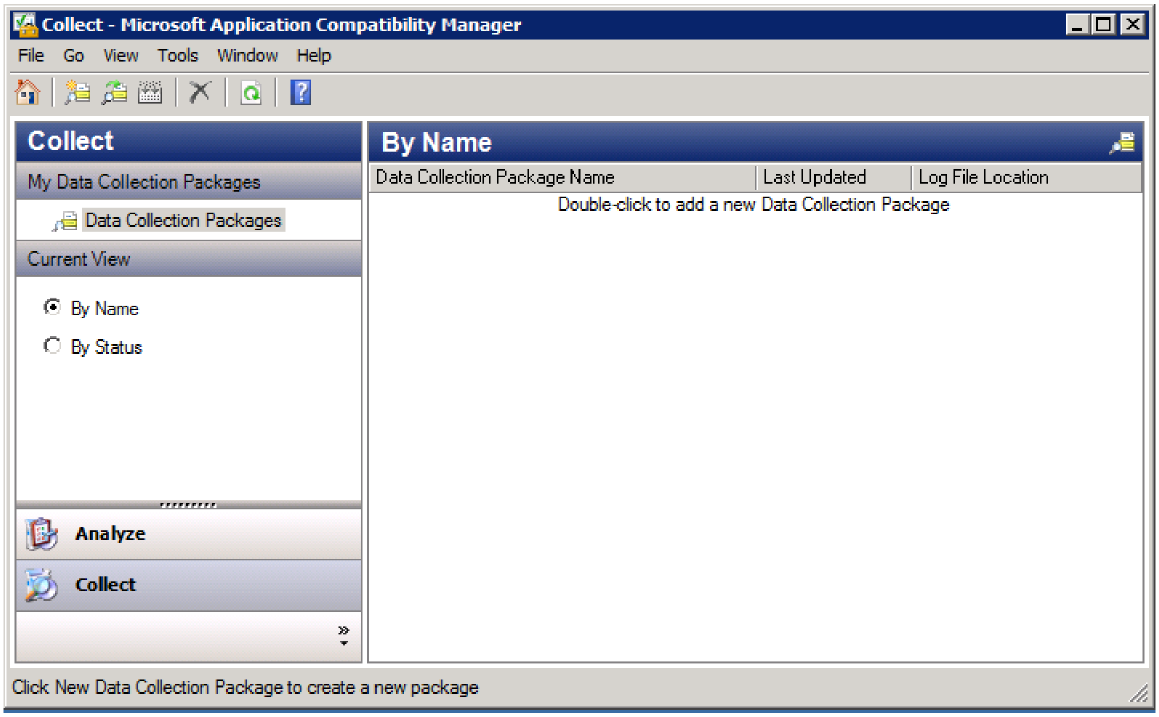
SQL Server: WDSSERVER\ACT

Database: MyActDatabase

Figure 6.10: Configuring the ACT database.

In a following screen, you'll be asked for your share path and name. Be aware that the Domain Computers group must be given write access to this location. The configuration wizard's final screen will ask for account credentials for the ACT Log Processing service. You may select either Local System or a domain user account.

You'll notice as the ACT launches that there's not much to see at first. Figure 6.11 shows ACT's Collect view. It is in this view where Data Collection Packages are configured and generated. Those packages are essentially a software installation in .MSI format that installs silently. Inside the package is the necessary logic that will evaluate computers, gather data, and report back application and compatibility information to the file share.



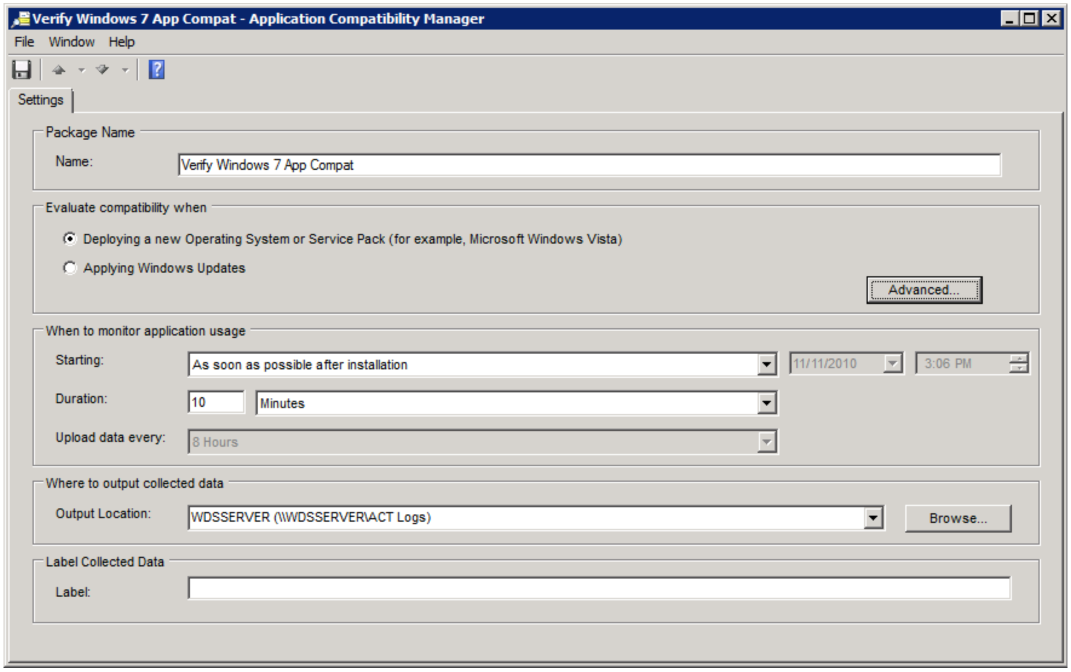
Double-click to add a new Data Collection Package

Click New Data Collection Package to create a new package

Figure 6.11: The ACT's console.

Creating a Data Collection Package

Creating a Data Collection Package begins by double-clicking the white space inside the ACT's right pane. Doing so brings forward a wizard screen similar to what you see in Figure 6.12. Each package's .MSI file can be installed by double-clicking it on a candidate system or by distributing it through a software delivery system. For this example, I'll create a package and simply double-click it on my WDS server, \\wdsserver.



Verify Windows 7 App Compat - Application Compatibility Manager

Package Name

Name: Verify Windows 7 App Compat

Evaluate compatibility when

● Deploying a new Operating System or Service Pack (for example, Microsoft Windows Vista)

○ Applying Windows Updates

When to monitor application usage

Starting: As soon as possible after installation 11/11/2010 3:06 PM

Duration: 10 Minutes

Upload data every: 8 Hours

Where to output collected data

Output Location: WDSSERVER (\\WDSSERVER\ACT Logs)

Label Collected Data

Label:

Figure 6.12: Creating a Data Collection Package.

I've named this package *Verify Windows 7 App Compat*, as you can see in Figure 6.12. I've also set the package to begin monitoring application usage as soon as possible after installation. The package has a duration of 10 minutes and will output its data to the ACT's log location. When you click Advanced, you will see a secondary screen that allows you to enable or disable any of the three compatibility evaluators: inventory collector, User account control compatibility evaluator, or Windows compatibility evaluators.

Choose File | Save and Create Data Collection Package to complete this step and ready the package for deployment. This process creates that .MSI file in the location you specify. Once created, double-click that .MSI file on any computer to begin collection, or deploy it through your software deployment solution.

This ACT Thing Seems Hard

You might be asking why this process seems somewhat more complicated than the MAP's inventory process. Recall that although WMI can be remotely queried over the network, it can only provide a limited set of data about installed applications. Some applications may not be exposed properly or completely within WMI. Thus, some of your applications simply won't be in the MAP's list.

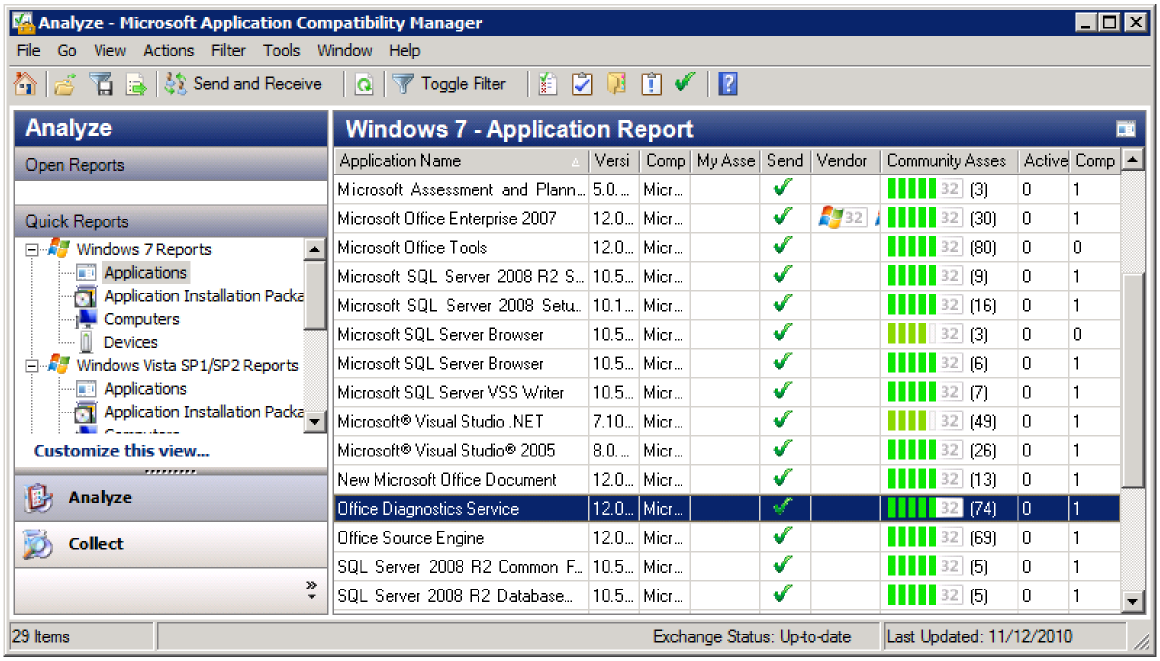
The ACT is intended to scale to thousands of computers. It is also designed to collect and report on other data, with that reporting to occur over a distributed amount of time to prevent abusing your file share. For all of these reasons, Microsoft uses an installable "agent" to accomplish these tasks. In the end, it is more work to get the data you need, but the data you'll get will be of much higher quality.

After letting the agent run for 10 minutes or so, you should begin seeing data in the log location's \Processed folder. When you do, return to the ACT and take a look at its Analyze view. Within the Analyze view, you'll find information about the applications the agents have found on your computers.

Analyzing and Keeping Track of Results

Now, here's where the exciting part happens. While in that view, click Send and Receive. Clicking this button shares your application information with Microsoft, while at the same time downloading Microsoft's corresponding compatibility information. You'll have an opportunity to see what data is sent to Microsoft before sending (it isn't much, but you'll want to confirm this sharing with your corporate security policies).

Once the synchronization is complete, the ACT's report will update to include information collected from the IT community about your applications. You can see in Figure 6.13 that the community has rated most of the applications on my \\wdsserver computer fairly well, with a slightly lesser rating for *Microsoft Visual Studio .NET* and the *Microsoft SQL Server Browser*. You should know that although the software seen in Figure 6.13 is all Microsoft software, this report will provide information on other software as well.



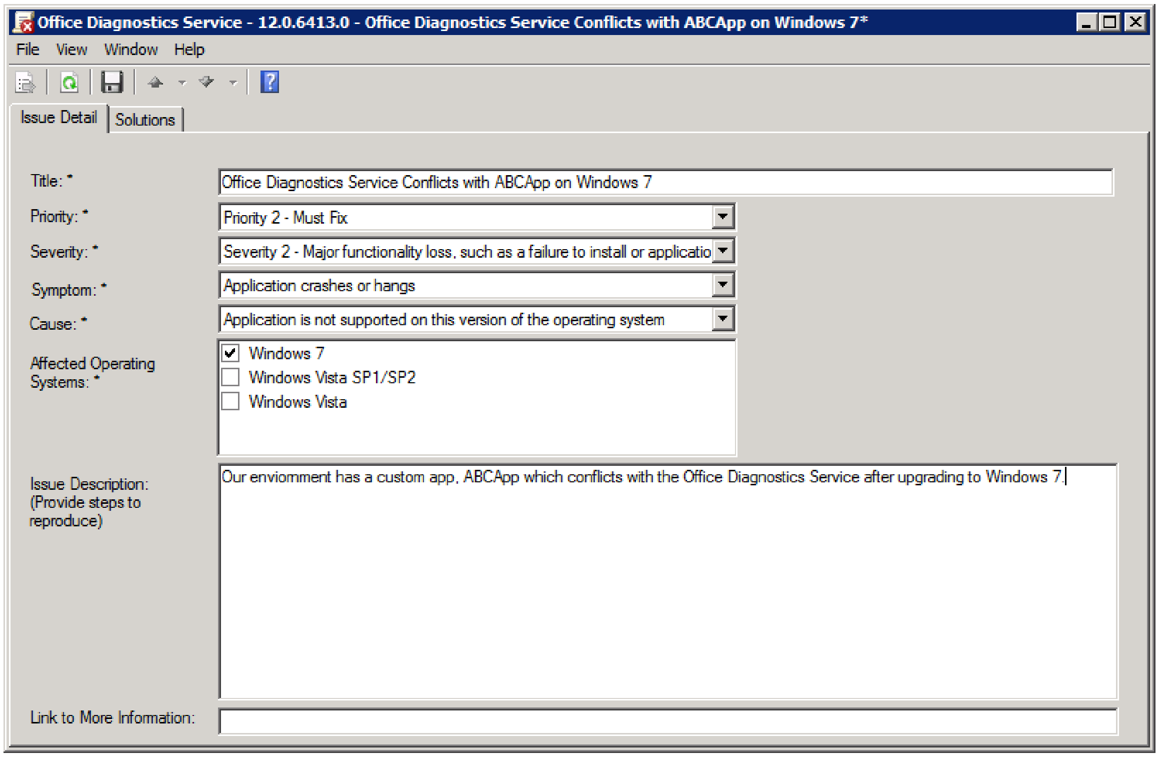
Windows 7 - Application Report

Application Name	Versi	Comp	My Asse	Send	Vendor	Community Asses	Active	Comp
Microsoft Assessment and Plann...	5.0. ...	Micr...		✓		32 (3)	0	1
Microsoft Office Enterprise 2007	12.0...	Micr...		✓	32	32 (30)	0	1
Microsoft Office Tools	12.0...	Micr...		✓		32 (80)	0	0
Microsoft SQL Server 2008 R2 S...	10.5...	Micr...		✓		32 (9)	0	1
Microsoft SQL Server 2008 Setu...	10.1...	Micr...		✓		32 (16)	0	1
Microsoft SQL Server Browser	10.5...	Micr...		✓		32 (3)	0	0
Microsoft SQL Server Browser	10.5...	Micr...		✓		32 (6)	0	1
Microsoft SQL Server VSS Writer	10.5...	Micr...		✓		32 (7)	0	1
Microsoft® Visual Studio .NET	7.10...	Micr...		✓		32 (49)	0	1
Microsoft® Visual Studio® 2005	8.0. ...	Micr...		✓		32 (26)	0	1
New Microsoft Office Document	12.0...	Micr...		✓		32 (13)	0	1
Office Diagnostics Service	12.0...	Micr...		✓		32 (74)	0	1
Office Source Engine	12.0...	Micr...		✓		32 (69)	0	1
SQL Server 2008 R2 Common F...	10.5...	Micr...		✓		32 (5)	0	1
SQL Server 2008 R2 Database...	10.5...	Micr...		✓		32 (5)	0	1

29 Items | Exchange Status: Up-to-date | Last Updated: 11/12/2010

Figure 6.13: The Windows 7 Application Report after synchronizing.

You can safely assume that information gathered in this report is equivalent to what you would be seeing on Microsoft's Windows 7 Compatibility Center Web site. That's a good thing. What it means is that the ACT's Application Compatibility Manager is a kind of workflow management solution for keeping track of all your applications. As you click through its settings, you'll notice that you can set your own assessment of each application as well as its deployment status, category, and priority. You can also document issues with applications and corresponding solutions. Figure 6.14 shows an issue with the Office Diagnostics Service conflicting with a custom application once upgraded to Windows 7.



Office Diagnostics Service - 12.0.6413.0 - Office Diagnostics Service Conflicts with ABCApp on Windows 7*

Title: * Office Diagnostics Service Conflicts with ABCApp on Windows 7

Priority: * Priority 2 - Must Fix

Severity: * Severity 2 - Major functionality loss, such as a failure to install or applicatio

Symptom: * Application crashes or hangs

Cause: * Application is not supported on this version of the operating system

Affected Operating Systems: *
☑ Windows 7
☐ Windows Vista SP1/SP2
☐ Windows Vista

Issue Description: (Provide steps to reproduce) Our enviomment has a custom app, ABCApp which conflicts with the Office Diagnostics Service after upgrading to Windows 7.

Link to More Information:

Figure 6.14: Documenting an issue with an application.

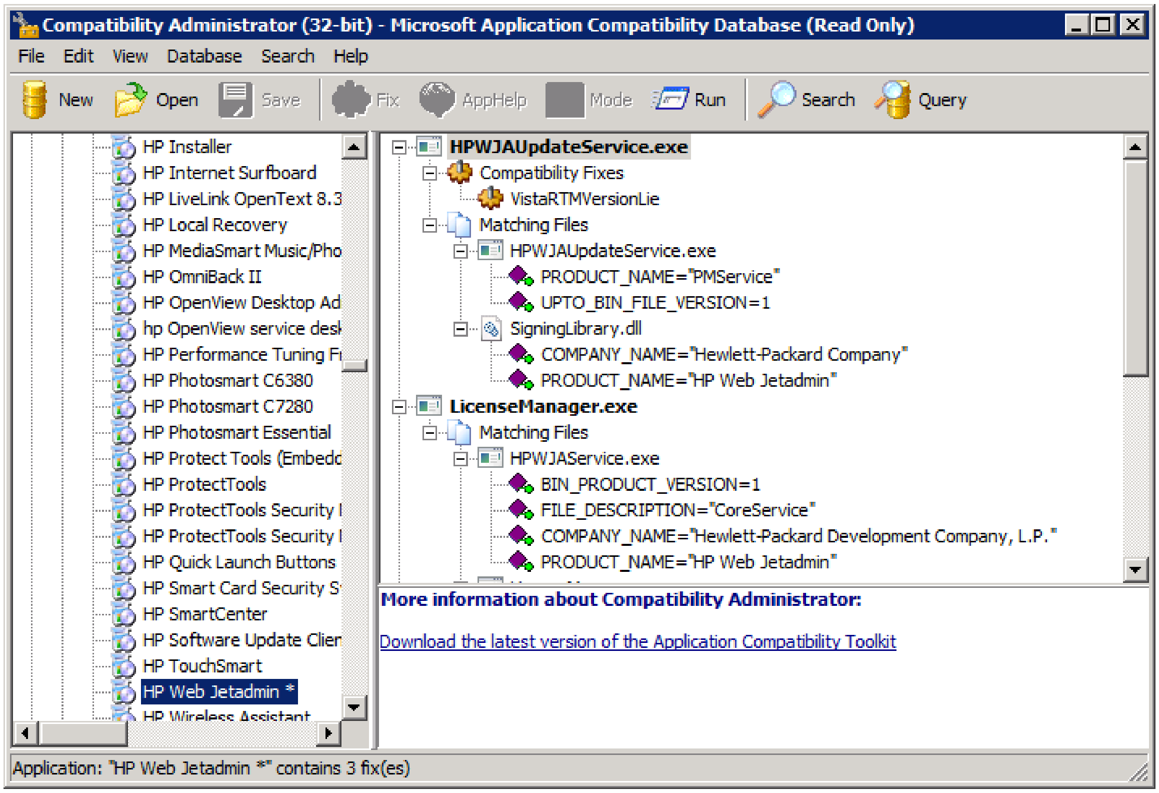
Note

The ACT is useful for more than verifying compatibility with Windows 7. It can also be used for verifying compatibility with Microsoft updates prior to deploying them with WSUS.

Fixing Incompatible Applications

All of this setup merely gets you to the point where you can begin analyzing and tracking your applications for incompatibility. ACT also provides a set of tools for actually fixing incompatible applications. One tool that you'll use in modifying applications to help them run is the ACT's Compatibility Administrator. Inside the Compatibility Administrator are more than 6500 known applications along with their accompanying fixes.

Two versions of the Compatibility Administrator are available, one each for 32-bit and 64-bit applications. In either, click the Applications node to expand the list of known applications. As you'll quickly find, fixes are available for many known applications, such as the HP Web Jetadmin application that Figure 6.15 shows. There you can see that the VistaRTMVersionLie fix can resolve an issue with the HPWJAUpdateService.exe file.



```
HPWJAUpdateService.exe
  Compatibility Fixes
    VistaRTMVersionLie
  Matching Files
    HPWJAUpdateService.exe
      PRODUCT_NAME="PMService"
      UPTO_BIN_FILE_VERSION=1
    SigningLibrary.dll
      COMPANY_NAME="Hewlett-Packard Company"
      PRODUCT_NAME="HP Web Jetadmin"
LicenseManager.exe
  Matching Files
    HPWJAService.exe
      BIN_PRODUCT_VERSION=1
      FILE_DESCRIPTION="CoreService"
      COMPANY_NAME="Hewlett-Packard Development Company, L.P."
      PRODUCT_NAME="HP Web Jetadmin"
```

More information about Compatibility Administrator:

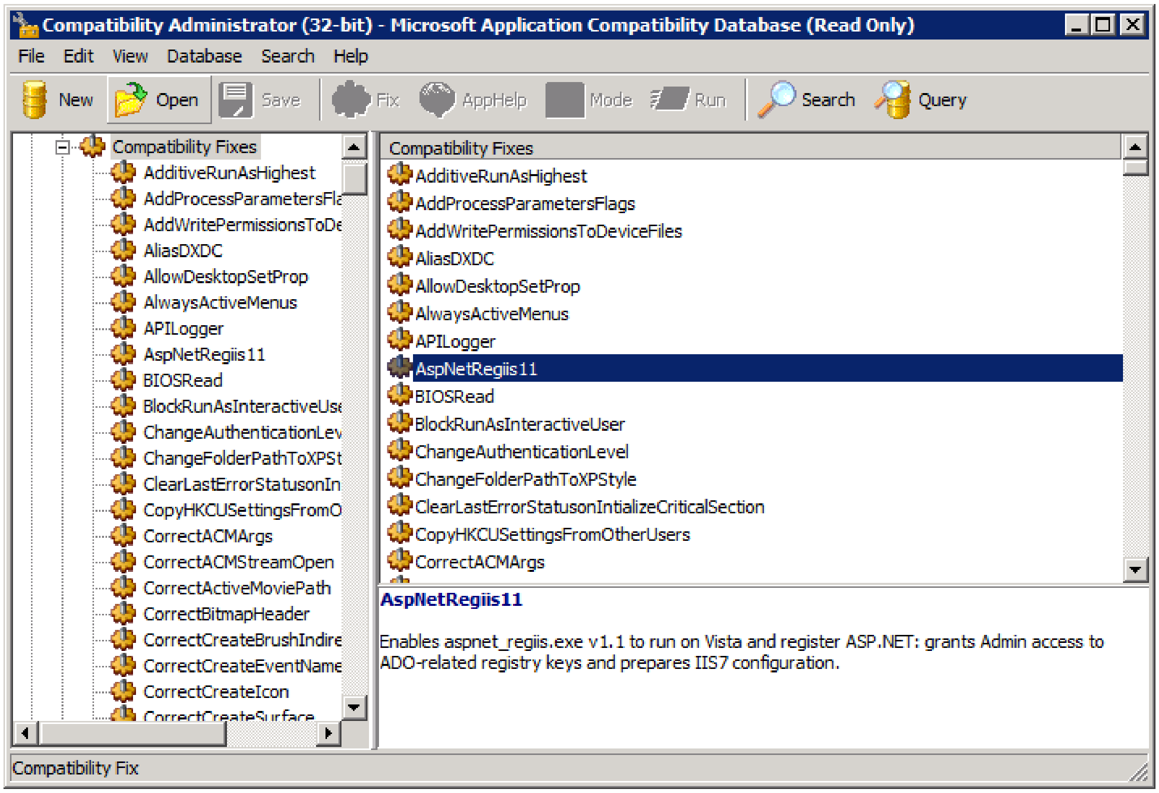
Download the latest version of the Application Compatibility Toolkit

Application: "HP Web Jetadmin *" contains 3 fix(es)

Figure 6.15: Compatibility Administrator.

Although this information is useful if your applications are in the list, many applications won't be available. Any custom and uncommon applications your company uses aren't likely to be present.

For applications not in the list, you'll need to test which compatibility fixes might resolve the problem. Built into the ACT are more than 360 compatibility fixes (sometimes called *shims*) that can be integrated into an application to return it to functionality. These fixes, a list of which can be seen in Figure 6.16, resolve issues with User Account Control, permissions, file virtualization and repathing, and numerous others.



Compatibility Fixes

- AdditiveRunAsHighest
- AddProcessParametersFlags
- AddWritePermissionsToDeviceFiles
- AliasDXDC
- AllowDesktopSetProp
- AlwaysActiveMenus
- APILogger
- AspNetRegiis11
- BIOSRead
- BlockRunAsInteractiveUser
- ChangeAuthenticationLevel
- ChangeFolderPathToXPStyle
- ClearLastErrorStatusonIntializeCriticalSection
- CopyHKCUSettingsFromOtherUsers
- CorrectACMArgs

AspNetRegiis11

Enables aspnet_regiis.exe v1.1 to run on Vista and register ASP.NET: grants Admin access to ADO-related registry keys and prepares IIS7 configuration.

Figure 6.16: The ACT's compatibility fixes.

What Fixes Do What?

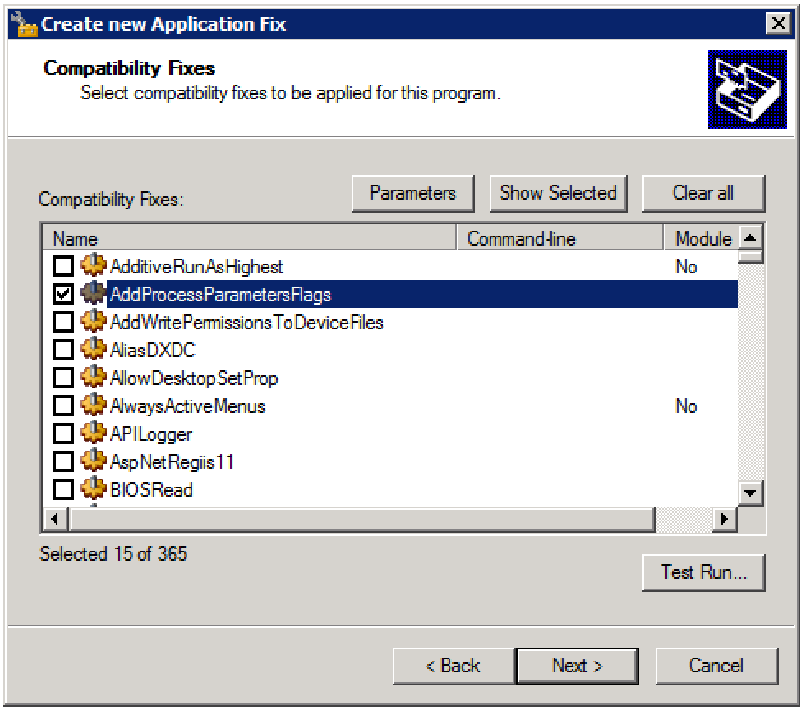
With 360 possible compatibility fixes to choose from, how do you know where to start? Good question, because the answer isn't immediately obvious. Check out the TechNet Web site at http://technet.microsoft.com/en-us/library/cc722305(WS.10).aspx. In this location, you'll find descriptions of each fix, along with symptoms that suggest when the fix might be applied.

Even with the symptoms and fix descriptions on this Web site, finding the fix that actually works will be a guess-and-check exercise. Problematic applications will require substantial work to see which fix (if any) will work. The ACT exists to create a singular database where discovered fixes can be logged so that you can measure your progress over time and document your findings.

To run tests against applications, you'll want to install the Compatibility Administrator to a reference computer that is running Windows 7. On that computer, install the application and verify that it is incompatible. Take careful note of exactly how that application is failing as well as error messages it gives or other notable behaviors.

To test a fix, create a Custom Database in the Compatibility Administrator on the Windows 7 computer. Right-click that database, and create a new Application Fix. Doing so will launch the Create new Application Fix wizard.

In this wizard, you'll be asked for information about the application, including its program file location. You'll also assign potential fixes to the application. Using this wizard enables you to tag applications with potential fixes until you find the correct set that works. Figure 6.17 shows one of the wizard's pages where compatibility fixes are tagged to an application. You can click Test Run to test the execution of that application after fixes are applied. The goal here is to verify whether assigned fixes indeed resolve an incompatibility.



Create new Application Fix

Compatibility Fixes

Select compatibility fixes to be applied for this program.

Compatibility Fixes:

Name	Command-line	Module
□ AdditiveRunAsHighest		No
☑ AddProcessParametersFlags		
□ AddWritePermissionsToDeviceFiles		
□ AliasDXDC		
□ AllowDesktopSetProp		
□ AlwaysActiveMenus		No
□ APILogger		
□ AspNetRegiis11		
□ BIOSRead		

Selected 15 of 365

Figure 6.17: Tagging a compatibility fix to an application.

Note

One easy starting point is to try setting the Compatibility Mode for the application to Windows XP. This Compatibility Mode automatically adds a series of adjustments to the application's execution space that may quickly solve the incompatibility. If an adjustment to the Compatibility Mode does not resolve the problem, trying out specific fixes should be your next step.

After configuring the fix, save it to a location on the computer. The file will have an .SDB extension. Then right-click the custom database and choose Install to install the fix. Once installed, try re-launching the application to see whether the incompatibility is resolved. If not, iterate through the previous steps until you locate the set of fixes that work with your application. Make sure to right-click the custom database, and choose Uninstall to remove the fix before adding a new fix.

Once you've discovered the correct set of fixes your application needs, you'll want to deploy those fixes along with the deployment of your application. That deployment will be different based on how you deploy applications. You can do so by including the compatibility fix database in your deployed image, distributing it via a software distribution system, or including it in a logon script by calling the *sdbinst.exe* file. Each of these options is explained in greater detail in the Microsoft TechNet article at http://technet.microsoft.com/en-us/library/cc794691(WS.10).aspx.

Solving Incompatibility: Not Hard but Time-Consuming

Reading through this chapter, you're probably thinking to yourself, "This seems like a lot of work." In all honesty, it is. Not all applications work well atop Windows 7, although that number is growing every day as vendors realize the need to update their code. There will always, however, be a set of applications that will never run just right. For those, tools like the MAP and particularly the ACT give you a mechanism to track down and potentially resolve their incompatibilities. Getting those fixes just right will take a bit of effort.

In just thirteen steps, you've been introduced to nearly all the pieces you need to be successful with your Windows 7 deployment. You can now deploy images with the assurance that user data and now applications will run successfully atop the new OS.

What you're likely still looking for is a way to pull all these steps together into a cohesive solution. You're nearly there. If you've been following along as we've been building this solution together, you've already got most of the pieces connected that you need to fully automate the solution. Taking that last step and completing the solution is the topic for the next chapter.

Download Additional eBooks from Realtime Nexus!