



Getting Started with Application Compatibility in a Windows Deployment

Disclaimer

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication. This white paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, AS TO THE INFORMATION IN THIS DOCUMENT. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation. Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property. Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place or event is intended or should be inferred.

© 2007 Microsoft Corporation. All rights reserved.

Microsoft, Windows and Windows Vista are either registered trademarks or trademarks of Microsoft Corporation in the United States or other countries or regions.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

Table of Contents

Introduction	4
Understanding the Application Compatibility Problem	4
Gathering an Application Inventory	4
Conducting a Manual Inventory	5
Automating the Inventory Collection	5
Deciding How Much to Inventory	6
Analyzing Application Compatibility Data	6
Identifying Business Critical Applications.....	7
Identifying High Priority Applications	7
Identifying Applications for Specific Roles	8
Next Steps.....	8
Resources	9

Introduction

If you have worked on an operating system deployment project before, you already know that application compatibility can be one of the most serious blocking issues. If you are working on an operating system deployment project for the first time, the scope of the work needed might come as a surprise. But, with proper planning, the project becomes easily manageable.

You can manage application compatibility proactively by keeping an accurate list of all programs in use in your organization, creating an “application portfolio.” Whether your organization has a managed environment complete with inventoried applications, a full-time application management team, and a test lab, or is approaching the issue of application compatibility for the first time, you may find it difficult to know where to start. This white paper explains the steps you should take to prepare for application compatibility testing and evaluation during a Windows Vista® operating system deployment project. Although the steps in this paper apply to a Windows Vista deployment, the concepts will translate well to your application lifecycle management.

After reading this paper, you should have a clear idea of where to start in the process of evaluating the impact of application compatibility in your deployment project. You’ll learn how to make an inventory, prioritize applications, and identify which applications are to be included in a specific role-based deployment image.

For greater depth of information, including step-by-step procedures, see Microsoft Deployment, the next version of Business Desktop Deployment (BDD) 2007 at <http://go.microsoft.com/?linkid=7730130>

Understanding the Application Compatibility Problem

Organizations typically have hundreds, if not thousands, of applications installed on all the computers in the network environment. This includes multiple versions of the same application, and even different updates applied to a single application version. Many of the applications and versions will be compatible with Windows Vista, but some may have compatibility issues with Windows Vista or with other installed applications or device drivers. The problems resulting from these incompatibilities can range from losing functionality in a single application (such as no longer being able to use a single command on a particular menu in the program) to critical kernel-level faults resulting in a stop error.

Where do you start the process of evaluating the application compatibility challenges your Windows Vista deployment may face? Best practices show that a successful process will include the following steps:

1. Collect information about current applications.
2. Prioritize and rationalize the applications to test and support.
3. Test the applications.
4. Mitigate Issues (remediate, upgrade, migrate, retire).

This document describes the first two points on the list: Collect information about your current inventory, and determine how you can begin to prioritize and rationalize your application inventory. The reference section at the end of this document provides links to in-depth information on tools, testing, issue remediation, and deployment in Microsoft Deployment Solution Accelerator.

Gathering an Application Inventory

The process of gathering an application inventory for your organization will depend on many factors, including:

Managed or unmanaged environment—Managed environments are much easier to inventory, because they control which applications can be installed on any managed computer. A managed environment will most likely already have a list of supported and approved applications, and you can view this list as complete for all managed computers. An unmanaged environment is challenging due to the complex task of discovering all the applications installed, both officially supported and approved applications, and the programs that users have installed on their own. Most organizations fall somewhere in the middle of this range: they have a combination of a central repository of applications and images, and a number of applications that are exceptions to the standard throughout the organization.

Centralized or Autonomous IT—Organizations with a centralized IT infrastructure will have an advantage in gathering an inventory of applications because they will be aware of, and in contact with, all of the departments within the organization. Decentralized IT infrastructures have a more complex task of communicating their individual inventories across organizational, cost center, and geographical boundaries.

Available Inventory Tools—Organizations that use a software management tool such as Microsoft Asset Inventory Service (a tool available through Microsoft® Desktop Optimization Pack for Software Assurance), Microsoft System Center Configuration Manager 2007, or Microsoft Systems Management Server 2003 (SMS) may already have a comprehensive list of applications present in the organization. Smaller organizations without management software will need to select a tool to gather the inventory. Although it is possible to take an inventory manually, you will run the risk of making mistakes and omissions. An inventory tool can effectively gather the list of applications running on every computer. The Application Compatibility Toolkit 5.0 includes an inventory tool that can help you automate this process.

Scope—The number of computers is a factor when gathering an inventory, but it may not be as obvious as simply specifying the number of computers to inventory. You can reduce the number of computers to be inventoried if they are managed (users can not install their own software) and if you can identify all the roles within the organization. In this case, you can inventory a sampling of computers from each specific role and location to gather an inventory. In lightly managed or unmanaged environments, users may be permitted to install their own software, and there may be hundreds or thousands of applications beyond the approved suite of programs. The number of user roles will also affect the total scope of the inventory, because you must include representative computers from each role to ensure proper coverage.

Conducting a Manual Inventory

A very small organization could choose to conduct a manual inventory. If you manage the IT infrastructure for a small organization, you might find it simplest to create a spreadsheet listing the applications on each computer. You can gather the list by logging on to each computer and writing them down from the Start menu, or by conducting a survey where each user can provide the list to you. The accuracy of these methods will be impacted by applications that are installed on a per-user basis and not visible when you log on, and by the thoroughness of the user's response.

Larger organizations would expand these techniques to include interviewing representatives from each department to help the application compatibility team create a list of applications used by that department. These interviews are also important for identifying the importance of applications within each role.

Obviously, a manual process doesn't sound like fun for anyone involved, but the greater risk is the possibility of making omissions and mistakes. If you consider that a highly effective automated inventory tool is available free from Microsoft in the Application Compatibility Toolkit ([ACT](#)), it makes more sense to automate this process.

Automating the Inventory Collection

In a well managed environment, you would always have an up-to-date list of installed applications for your organization. You can accomplish this by implementing a software management or inventory tool and

using it on a regular basis. A system management tool such as Configuration Manager 2007 or SMS 2003 can assist by maintaining a list of applications that are approved and installed in the organization. But if you currently work in a lightly managed or unmanaged environment, you probably do not have an up-to-date inventory of all the applications in your corporate environment. In this case, you may choose to use the Application Compatibility Toolkit (ACT), which can be downloaded from Microsoft. The ACT contains an inventory tool among other testing and remediation tools, which can be used to automate the collection of application inventories for all computers in your organization or for a subset, based on how you choose to deploy the ACT inventory client. ACT is a valuable tool for both managed and unmanaged environments, regardless of size. For more information on ACT, see Microsoft Application Compatibility Toolkit 5.0 at <http://go.microsoft.com/?linkid=7730132>

Deciding How Much to Inventory

Small organizations have an easy answer to the question of “How many computers should I inventory?” In a small organization, you could simply inventory every computer on the network. Large organizations, on the other hand, may face the task of inventorying thousands of computers, and those computers may be spread over multiple time zones and geographic locations.

In well-managed environments, you can base your inventory on one computer for each role and supported configuration. In lightly managed environments, you will need to determine what the typical configurations are for each role, and inventory at least one computer for each role, while understanding that users will have added applications and tools. If there is any variation in the various configurations, you will need to be sure to capture inventory information for at least one of each configuration. By using an automated inventory tool, you have the option of inventorying every computer, which would provide the most complete view of your application portfolio.

Analyzing Application Compatibility Data

Before you can move from the inventory phase into actual testing of your applications, you will need to make some hard decisions regarding application priority and about which applications and versions will be officially supported after the deployment is complete.

The first review of the complete inventory is fairly straightforward:

Application redundancy—Your organization has more than one application performing the same tasks.

Application relevancy—Your organization has multiple versions of an application including outdated and unsupported versions. Depending on the application and your infrastructure, you may want to check on application-to-application dependencies before moving to an entirely new version of a particular application.

Application necessity—Your organization has applications that are irrelevant to the day-to-day work being done in your organization.

You can use the application inventory to reduce application redundancy. For example, your organization might have multiple applications used to create graphics. By selecting a single application, and a single version of that application as a standard, your organization can save money spent on support. If you identify a version of your required application that is proven compatible with the version of Windows® operating system you are deploying, you can also reduce deployment testing by focusing on this single application and version.

The next step is to categorize the applications into groups based on the relative importance of that application within your organization. To understand this, you will need to work with the business owners of each application. Some applications are considered business critical, such that your business will halt if the application fails or is unavailable. Other applications are very important, but there are ways to keep some business going if they fail. Although the individual terminology may vary, here are some common priority levels to consider:

Business Critical—An application that stops your organization completely if it fails. If in doubt as to whether an application falls into this category, consider whether the organization can make any money while the application is unavailable, and whether personnel can continue working at all without the application. If this application fails in the middle of the night, how long it would be until your pager goes off? Business critical applications often have service level agreements (SLAs) that state that support staff must respond to a failure within 15 minutes or less.

High Priority—Applications that perform a vital role in a department or across an organization. A failure in a high priority application may disable a department or a single business function in the organization. If a high priority application fails, the organization can continue doing business, but one or more departments may be seriously hindered. A typical SLA for a high priority application would be measured in hours.

Important—An important application that is used frequently, but will not cause a stoppage of work if it fails. An example would be a spreadsheet or word processor application that is widely used, but not related to a fundamental business function. SLAs for important applications may be measured in days.

Optional—Approved applications that are in limited use and not directly related to a business function. An application in this category is usually not covered by an SLA, and support would be considered “best-effort.”

Assigning priority levels to applications is often a subjective process, and may be subject to periodic revision. This is a valid reason to maintain a team or committee to monitor the application portfolio even between operating system deployments.

After you have prioritized your application inventory, you should also take some time to identify applications that can be eliminated. Removing old versions of applications can greatly reduce the amount of application testing that must be performed prior to deploying a new operating system. Consider whether the older versions can still be supported by the manufacturer. Gathering an application inventory is a good opportunity to check for improper license use, such as in cases where the application is installed on too many computers.

Identifying Business Critical Applications

Some business critical applications are easy to identify and some are not. If your organization does business through an Internet presence, word processing, image manipulation, and page coding applications might be considered business critical. Business critical categorization will vary by job role, too. To a call center, an application that monitors server stability might be considered critical, although other departments would place no value on it at all.

If your organization has a disaster recovery plan (DRP) or a business continuance plan (BCP) in place, the business critical applications should already be identified in those plans.

Some criteria for identifying business critical applications include:

- The application represents a primary function of the organization such word processing or page design software to a publishing company.
- The organization will suffer a significant financial impact if this application fails.
- If this application fails during off-hours, someone’s pager will go off within minutes.

Identifying High Priority Applications

High priority applications may be easy to identify in your own department, but difficult in other departments. Sometimes you can identify a high priority application by the percentage of computers it is installed on, though this can be misleading. An application that was developed internally to monitor the amount of toner in the company’s laser printers might be deployed to every computer but actually used by

very few people. If your organization has a disaster recovery plan (DRP) or a business continuance plan (BCP) in place, the high priority applications may already be identified in those plans.

Some criteria for identifying high priority applications include:

- The application is used the majority of the work shift.
- The application is used by a large percentage of users within the organization.
- Without the application, it would be difficult for the organization or department to conduct business normally.
- Without the application, daily operations would be affected and there may be a measurable financial impact.

Identifying Applications for Specific Roles

The concept of user roles can be helpful in determining the components of specific operating system images for deployment. By separating your user base into specific roles, you can more easily define the applications to be layered on top of the operating system image. For example, if you define a user role of accounting, you can specify that, in addition to the operating system, the users in this role will receive Microsoft® Office and the organization's standard accounting software. Some possible roles might include:

- Human resources
- Information worker
- IT support desk
- Marketing
- Developer
- Executive

There are, of course, applications that are standard across all operating system images. Examples might include e-mail and antivirus applications. These applications will be part of a core image and should all be tested extensively for compatibility and stability with Windows Vista®. Deployment images for the other roles are based on the core image, adding other applications used by the role.

Deployment roles can also be prioritized to lower the effort required for testing application compatibility. Role-based configurations for critical operations should receive the most testing, and lower priority images can receive lower amounts of testing. Role configurations with only minor differences from the core image can receive minimal testing. This prioritization can be carried over to the actual deployment as well. Roles that contain applications without compatibility issues can be deployed sooner, and those with issues can be deployed later after testing and remediation have taken place.

Next Steps

The process outlined in this document should produce an application catalog for your organization that is prioritized in terms of testing required, specific user roles, and which applications will be deployed with Windows Vista®. This catalog gives you the raw information needed to progress to the next steps in the process. Using the information in your application catalog, your next steps should include:

- After you understand the scope of work involved in your organization, you will need to organize a team to work through both the technical and business decisions that need to be made. The size and scope of this team is dependent on the size and scope of your deployment. It may be that you are the project manager, test lead, and general go-to resource. Larger organizations will need a more formal team with well-defined roles. Guidance for putting together a team can be found in the Microsoft® Deployment Solution Accelerator guidance. You can find more information on creating deployment project teams in the Plan, Build, and Deploy Guide at <http://go.microsoft.com/?linkid=7730133>
- Decide which applications will be officially supported after the deployment. This gives you the option of placing some of the older, little-used applications in the category of unsupported applications. These programs typically receive “best-effort” support from your IT department with no assurance of compatibility.
- Create a test plan for the applications that will require testing. You should plan to test your role-based images as well, because some application compatibility issues may only emerge in combination with other applications.
- Create a test lab in which to perform application compatibility testing. Very often, this will be the same lab used to perform testing on the deployment images. Consider using virtualization software such as Microsoft Virtual PC 2007 or Microsoft Virtual Server 2005 as an alternative to dedicated computers for the bulk of application testing to reduce hardware cost.
- Identify applications that have compatibility issues and find the correct remediation. Test the remediation to ensure that the issues are actually corrected. The ACT contains information on compatibility issue remediation in its documentation. You can also find more information in the Application Compatibility Feature Team Guide in Microsoft Deployment.
- Identify members of your pilot deployment test group. You should pick people that represent each major role within your organization so that you can get valuable feedback on each role-based image being deployed. Consider the place of upper management in the pilot group. Some executives that are technically savvy may elect to dedicate a computer to pilot testing.
- Sign off on application compatibility for the operating system deployment. Your organization may or may not employ a formal sign-off procedure to mark completion of each project milestone, but the application compatibility team’s involvement is just one part of the overall operating system deployment project.

Deploy the operating system and applications to the organization.

There may be other tasks involved as well, depending on your organization’s needs and culture. The decisions you make regarding the priority of applications and testing may be affected by the structure of your organization as well. For example, you may want to add a step to reevaluate the list of applications to be deployed with the operating system to see if there are new applications that should be added.

These further steps are discussed in greater depth in other documentation such as Microsoft Deployment for Windows Vista. See the Resources section below for further information.

Resources

[Microsoft Deployment Solution Accelerator](http://go.microsoft.com/?linkid=7730134)

<http://go.microsoft.com/?linkid=7730134>

Microsoft Deployment for Windows Vista® contains an extensive set of documentation on every phase of an operating system deployment, including application compatibility.

[Download ACT 5.0](http://go.microsoft.com/?linkid=7730132)

<http://go.microsoft.com/?linkid=7730132>

[Download the ACT 5.0 Deployment Guide](#)

<http://go.microsoft.com/?linkid=7730135>

[Review application compatibility content on TechNet and MSDN](#)

<http://go.microsoft.com/?linkid=7730136>

[Access the Windows Vista Application Compatibility Cookbook on MSDN](#)

<http://go.microsoft.com/?linkid=7730137>

[Enlist the assistance of Application Compatibility Factory \(ACF\) partners](#)

Several global System Integrators have partnered with Microsoft to offer deep application compatibility expertise to assist you with application testing and remediation. ACF partner services focus on custom line-of-business applications, and can scale to some independent software vendor applications. For more information on the ACF, see <http://go.microsoft.com/?linkid=7730138>