



Microsoft ACT: Phase 3 - Testing and Mitigating Issues

Microsoft Corporation

Published: May 2010

Abstract

This paper provides information about testing and mitigating the known compatibility issues by using the Microsoft® Application Compatibility Toolkit (ACT) 5.6 and its associated tools, which include the Internet Explorer Compatibility Test Tool (IECTT), the Standard User Analyzer (SUA), and the Compatibility Administrator.

This information applies to the following operating systems, with requirements specified by the ACT:

- Windows® 7
- Windows Vista®
- Windows Server® 2008 R2
- Windows Server 2003
- Microsoft Windows XP with Service Pack 2 (SP2)
- Microsoft Windows 2000

Microsoft®

Copyright Information

This document is provided “as-is”. Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it.

Some examples depicted herein are provided for illustration only and are fictitious. No real association or connection is intended or should be inferred.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

© 2010 Microsoft Corporation. All rights reserved.

Active Directory, Internet Explorer, Microsoft, Windows, Windows NT, Windows XP, Windows Vista, Windows 7, and Windows Server are trademarks of the Microsoft group of companies.

All other trademarks are property of their respective owners.

Contents

Microsoft ACT: Phase3 - Mitigating Issues by Using Compatibility Fixes	4
How ACT Works	4
General Testing Methodology	5
Creating the Test Environment	5
Test Lab Requirements	5
Determining How to Model the Production Environment	6
Configuring the Test Environment for Automated Testing	7
Determining When Virtualization Is Appropriate	7
Testing and Mitigating User Account Control-Related Issues with the Standard User Analyzer and the Compatibility Administrator	8
Collecting the UAC-related issues	8
Reviewing the SUA Tool Results	9
Applying the Recommended Mitigations	9
Testing the Mitigations	10
Creating a Custom Compatibility Fix Database for Deployment	10
Removing the SUA Mitigations Database	11
Deploying the Contoso.sdb Database to Your Environment	11
Testing Your Mitigation Packages	12
Reporting the Compatibility Mitigation Status to Stakeholders	13
Resolving Outstanding Compatibility Issues	13

Microsoft ACT: Phase3 - Mitigating Issues by Using Compatibility Fixes

After you analyze your issues in the Application Compatibility Manager, you can continue to explore your compatibility issues, by using several development tools provided with the Microsoft® Application Compatibility Toolkit (ACT) 5.6. The development tools enable you to test for a variety of compatibility issues, including Web site and Web application issues and issues related to running as a Standard User in Windows® 7. Additionally, ACT provides a tool that can help you resolve many of your compatibility issues, the Compatibility Administrator. To successfully resolve your compatibility problems, you must follow these steps.

1. **Identify your most critical applications.** Create an inventory of your organization's applications, and then verify certification status of the included applications to see whether they require testing.
2. **Identify any application compatibility problems.** Test each application, determining any compatibility issues if necessary.
3. **Resolve any application compatibility issues.** Identify and create application compatibility solutions, by using the ACT tools, which include the Internet Explorer Compatibility Test Tool (IECTT), the Standard User Analyzer (SUA), and the Compatibility Administrator.
4. **Deploy or distribute your test and certified applications and solutions.** Use a deployment and distribution tool, such as Microsoft System Center Configuration Manager 2007 to deploy your certified applications and compatibility issue solution packages to your client desktops.

For more detailed information and additional procedures that are available for any of the tools included in this white paper, see the associated topics in the ACT Help.

How ACT Works

ACT 5.6 provides a way for you to create an inventory for your organization, including your installed applications, computers, and devices. It also enables you to collect compatibility data, to determine the impact of that data in your organization, and, finally, to create mitigation packages to fix the compatibility issues, when possible. There are three phases for effectively using ACT in your organization. The three phases are:

- **Phase 1: Collecting Data.** Before you can analyze your potential compatibility issues, you must first collect your organization's inventory and the associated compatibility issues. For more information, see the "Microsoft ACT: Phase 1 – Collecting Data" white paper, available for download from the **Related Resources** section of the [Microsoft Application Compatibility Toolkit 5.6 Download](#) Web page.
- **Phase 2: Analyzing Issues.** After collecting your inventory and associated compatibility data, you can organize and analyze your issues. This includes categorizing, prioritizing,

Microsoft ACT: Phase3 - Testing and Mitigating Issues

setting your deployment status, and setting your application assessment to create customized reports. For more information, see the "Microsoft ACT: Phase 2 – Analyzing Issues" white paper, available for download from the **Related Resources** section of the [Microsoft Application Compatibility Toolkit 5.6 Download](#) Web page.

- **Phase 3: Mitigating Issues by Using Compatibility Fixes.** After analyzing your compatibility issue reports, you can do the following:
 - Manually test your applications for functionality-related issues
 - Use the Standard User Analyzer (SUA) tool to automatically test for User Account Control (UAC)-related issues
 - Create mitigation packages for valid issues by using the Compatibility Administrator

General Testing Methodology

When testing an application in a new operating system, we recommend that you retain the default security-feature selections. We also recommend that you thoroughly test the applications, replicating as many of the usage scenarios from within your organization as possible. Finally, we recommend that you enter your issues and solutions into the Application Compatibility Manager, so that you can track the data from a central location.

When testing a Web site or a Web application, we recommend that you include both intranet and extranet sites, prioritizing the list based on how critical the site or the application is to your organization. We also recommend that you thoroughly test the Web sites and Web applications, replicating as many of the usage scenarios from within your organization as possible.

Creating the Test Environment

Your test environment should be a long-term investment in the overall deployment process. Retain the test environment after the deployment to assist in future deployment projects. To create the test environment, you must:

- Determine how to model the production environment in the test environment.
- Configure the test environment to support automated testing of the mitigation strategies.

Test Lab Requirements

We recommend that you establish a dedicated and isolated lab environment for use in developing and testing the application-compatibility mitigation. The lab should mirror your production environment as closely as possible. In some cases, you might find that it is better to open the test network to existing production services, instead of replicating your production environment in detail. For example, you might want to permit your Dynamic Host Configuration Protocol (DHCP) packets to pass through routers into the test network.



Note

Microsoft ACT: Phase3 - Testing and Mitigating Issues

Some operations can be safely conducted in the production environment, such as the application inventory-collection process.

At a minimum, your lab environment should include:

- DHCP services
- Domain Name System (DNS) services
- Microsoft® SQL Server® 2008, Microsoft SQL Server 2005, or either version of the Microsoft SQL Server Express Edition
- Lab test user accounts, with both normal user and administrative privileges
- Network hardware to provide connectivity
- Internet access (for downloading updates, files, and so on)
- Test computers that accurately reflect production computers in both software and hardware configuration
- A software library representing all the applications to be tested
- Microsoft System Center Configuration Manager 2007 or Systems Management Server (SMS) 2003 with Service Pack 1 and the SMS Operating System Deployment (OSD) Feature Pack (optional)
- Microsoft Virtual Server 2005 and Virtual PC 2004 (optional)
- Windows® Internet Name Service (WINS) services (optional)

Determining How to Model the Production Environment

The goal of the test environment is to model your production environment. The more accurate the production environment, the greater the validity of the testing performed in that test environment.

We recommend the following best practices in creating your test environment.

- Use virtual or physical images of production computers to create their test environment counterparts. Virtual or physical images can help ensure that the test environment configuration accurately reflects the production environment. In addition, the images contain live information (such as users, user profiles, and file permissions) to use in testing.
- Physically separate your test environment from your production environment. A physically separate test environment enables you to use an identical IP configuration and helps ensure that tests conducted in the test environment do not affect the production environment. Using the identical IP address, subnets, and other network configuration information helps to ensure the fidelity of the test environment. However, duplicating IP addresses might not always be the best option when applications do not rely on a hard-coded IP address. You might also pass some network traffic through the router from the production environment to reduce the need for replicating network services. For example, opening the ports for DHCP to pass through eliminates the need for a separate DHCP server in the test lab.
- Ensure that your test environment is at the same service-pack and update level as your production environment. Before performing your tests, update your lab environment by applying service packs and updates, or by refreshing the virtual or physical images of your

Microsoft ACT: Phase3 - Testing and Mitigating Issues

production counterparts. Consider adding the test environment to the change-management process to simplify tracking the updates.

- Ensure that you perform all of your tests by using accounts that have similar permissions as the accounts in your production environment. For example, if your organization does not allow users to run as Administrators on their local computers, ensure that similar permissions are not granted to users in the test environment. This process ensures that you can determine potential security issues.

Configuring the Test Environment for Automated Testing

In most instances, you must test the mitigation strategies more than once and be able to revert reliably to a previous test state. Automating your testing process enables you to ensure reproducibility and consistency in your testing process.

- Using test automation tools enables you to run your test cases in a standardized, reproducible manner.
- Using disk-imaging software for physical images of the servers and using software virtualization features for reversing changes to virtualized hard disks enable you to restore your test environment back to a previous state.

Determining When Virtualization Is Appropriate

You can create and run virtualized servers by using Virtual PC 2004 for Windows Server® 2003, or Virtual Server 2005 R2 for Windows Vista® or Windows 7. There are both advantages and disadvantages to using virtualization, so you must determine whether this is right for your organization.

Advantages	Disadvantages
<ul style="list-style-type: none">• You can support a large number of servers in a limited amount of physical space. Virtualization enables you to run as many virtual servers as the physical computer's resources (such as memory, multiple processors, and free disk space) allow.• You can more easily share your test environment between teams. For example, your test team can create a virtualized test environment and then provide a copy to your development team for use in its development processes.• Virtualization enables multiple users to perform simultaneous testing, mimicking the ability for each user to have a dedicated	<ul style="list-style-type: none">• You might find that virtualized servers are slower than their physical counterparts. The performance of virtualized servers is significantly reduced, because many of the physical resources (such as disks) are virtualized.• You might find that some hardware-specific device drivers and applications are not supported in virtualized servers.

Advantages	Disadvantages
<p>test environment.</p> <ul style="list-style-type: none"> You can easily restore your environment to a previous state. For example, you can revert to a previous state by using the Undo Disks option. 	

Testing and Mitigating User Account Control-Related Issues with the Standard User Analyzer and the Compatibility Administrator

The Standard User Analyzer (SUA) tests your applications to detect potential compatibility issues related to User Account Control (UAC). SUA then recommends the appropriate mitigations to fix the issues. UAC, formerly known as Limited User Account (LUA), requires that all users (including members of the Administrator group) run as Standard Users until the application is deliberately elevated by using the security prompt dialog box. However, applications that require access and privileges for locations that are not available to a Standard User cannot run properly with the Standard User role.

In this section, we run the SUA tool on an application with known UAC issues and then apply the recommended mitigations.

Collecting the UAC-related issues

First, use the SUA tool to test an application and to locate the UAC-related issues.

► To collect the UAC-related issues by using SUA

1. Click **Start**, point to **All Programs**, point to **Microsoft Application Compatibility Toolkit 5.6**, point to **Developer and Tester Tools**, and then click **Standard User Analyzer**.

The SUA tool starts.

2. In the **Target Application** field, browse to the **\Program Files\Microsoft Application Compatibility Toolkit\Compatibility Administrator (32-bit)\Demo Application\StockViewer** directory, and then double-click **StockViewer.exe**.

3. Clear the **Elevate** check box, and then click **Launch**.

The StockViewer tool attempts to start and the **Permission denied** dialog box appears.

4. Click **OK**.

The StockViewer application starts, despite the warning.

5. On the toolbar, click the **Trends** button.

An error message appears, stating that the application cannot create the object.

Microsoft ACT: Phase3 - Testing and Mitigating Issues

6. Click **OK** to close the error message.
7. On the **Tools** menu, click **Options**.
An error message appears, stating that there was an unhandled exception in the application.
8. Click **Continue** to close the error message and to continue testing the application.
9. On the **Help** menu, click **Check for Updates**.
Another error message appears, stating that there was an unhandled exception in the application.
10. Click **Continue** to close the error message and to continue testing the application.
11. Close the StockViewer application.
SUA automatically collects and shows the data that was found during the application run time.

Reviewing the SUA Tool Results

After collecting UAC-related issues from the StockViewer tool, review the information in the SUA tool.

To review the SUA tool results

1. In the SUA tool, click the **Registry** tab to review the registry issues.
2. Click the **Token** tab to review the issue that occurred when checking the token.
3. Click the **Name Space** tab to review the issues that occurred with the global name space.
4. Click the **Other Objects** tab to review the UAC-related mapping issue that occurred.
5. Click the **Process** tab to review the issues that occurred while launching a process.

Applying the Recommended Mitigations

After reviewing the UAC-related information from the various tabs of the SUA tool, you can review and apply the recommended mitigations.

To review and apply the recommended mitigations

1. In the SUA tool, on the **Mitigation** menu, click **Apply Mitigations**.
The **Mitigate AppCompat Issues** dialog box appears.
2. Review the recommended compatibility fixes. For more detailed information about each compatibility fix, see **Windows Vista and Newer Operating Systems** or the **Windows XP and Earlier** topics.
3. Click **Apply**.
The SUA tool generates a custom compatibility fix database and automatically applies it to your local computer, so that you can test the fixes to see if they worked.

Testing the Mitigations

After the SUA tool applies the custom compatibility fix database to your computer, you must test the StockViewer application to see if all of your issues are fixed.

► To test the recommended mitigations

1. Browse to the **\Program Files\Microsoft Application Compatibility Toolkit\Compatibility Administrator (32-bit)\Demo Application\StockViewer**, and double-click **StockViewer.exe**.
The StockViewer application starts without causing the **Permission denied** dialog box to appear.
2. On the toolbar, click the **Trends** button.
The **Trends** screen appears.
3. On the **Tools** menu, click **Options**.
The **Options** page appears.
4. On the **Help** menu, click **Check for Updates**.
The StockViewer tool successfully checks for updates.

Creating a Custom Compatibility Fix Database for Deployment

Now that you know the recommended compatibility fixes work, you can create your own custom compatibility fix database to deploy across your organization.

► To create a custom compatibility database

1. Click **Start**, point to **All Programs**, point to **Microsoft Application Compatibility Toolkit 5.6**, point to **Developer and Tester Tools**, and then click **Compatibility Administrator**.
The Compatibility Administrator appears.
2. Select the **New Database (1) [Untitled_1]** custom database.
3. On the toolbar, click the **Fix** button.
The **Create New Application Fix** wizard appears, showing the **Program information** page.
4. In the **Name of the program to be fixed** field, type **StockViewer**.
5. In the **Name of the vendor for this program** field, type **Contoso**.
6. In the **Program file location** field, type **C:\Program Files\Microsoft Application Compatibility Toolkit\Compatibility Administrator (32-bit)\Demo Application\StockViewer\StockViewer.exe**, and then click **Next**.
The **Create new Application Fix** wizard shows the available Compatibility Modes.
7. Leave the fields blank, and then click **Next**.
The **Create new Application Fix** wizard shows the available Compatibility Fixes.

Microsoft ACT: Phase3 - Testing and Mitigating Issues

8. Check the boxes for the **ForceAdminAccess**, **LocalMappedObject**, **VirtualizeHKCRLite**, and **ElevateCreateProcess** compatibility fixes, and then click **Next**.

The **Create new Application Fix** wizard shows the available matching information.

9. Accept the default matching information, and then click **Finish**.

The custom database is created in the **Custom Databases** node.

10. Select the new custom database.

11. On the toolbar, click **Save**

12. In the **File Name** field, type **Contoso**, and then save the file, contoso.sdb, to your desktop.

Removing the SUA Mitigations Database

Before you deploy your custom compatibility fix database, remove the original database created by the SUA tool and then install and test the Contoso.sdb database again to ensure that the compatibility fixes were successfully applied to your local computer.

▶ To remove the SUA Mitigations database

- In the Compatibility Administrator, expand the **Installed Databases** node, right-click the SUA Mitigations database (the only installed database that includes the StockViewer application), and then click **Uninstall**.

The SUA Mitigations database is uninstalled from your local computer.

▶ To install the custom database

1. In the **Custom Databases** node, right-click the Contoso.sdb database, and then click **Install**.
2. Verify that the Contoso database appears in the **Installed Databases** node.
3. Repeat the steps listed in the Testing the Mitigations section of this topic to ensure that the fixes have been successfully applied to the computer.

Deploying the Contoso.sdb Database to Your Environment

Finally, you can deploy the Contoso.sdb database into your environment so that the StockViewer application functions properly on all of your computers. For the purpose of this exercise, we simulate placing the Contoso.sdb database on a network share, and then we create a script to enable a management tool, such as Microsoft® System Center Configuration Manager, to install the database for later deployment to your environment.

▶ To create the network share location

1. Create a new folder on your desktop, named **SDBShare**.

Microsoft ACT: Phase3 - Testing and Mitigating Issues

2. Right-click the SDBShare folder, and then click **Properties**.
The **Properties** dialog box appears.
3. Click the **Sharing** tab, and then click **Advanced Sharing**.
4. Select the **Share this folder** check box, accept the default values, and then click **OK**.
5. Close the **Properties** dialog box.
6. Drag the Contoso.sdb database file, previously created and stored on your desktop, into the SDBShare folder.

The compatibility fix database is now available for installation by remote computers.

► To create the deployment script

1. To retrieve your computer name, open a Command Prompt window and type `net config workstation`.
2. Click **Start**, point to **All Programs**, click **Accessories**, and then click **Notepad**.
The Notepad tool appears.
3. Type `sdbinst "\\<your_computer_name>\SDBShare\Contoso.sdb" -q`, and then save the file to your desktop as SDBInstScript.cmd.
4. Click **Start**, point to **All Programs**, click **Accessories**, right-click **Command Prompt**, and then click **Run as Administrator**.

A Command Prompt window appears with Administrator permissions.

5. Type `C:\users\<your_profile_name>\desktop\sdbinstscript.cmd`, and then press **ENTER**.

The Contoso.sdb database file is installed to the local computer. Run this script on each computer in your environment to ensure that the applications run properly.

Testing Your Mitigation Packages

To test your mitigation packages, you run a series of tests in the test environment and one or more pilot deployments in the production environment.

► To test your mitigation packages

1. Perform the following steps for each of the applications for which you have developed mitigations.
 - a. Test the mitigation strategy in your test environment.
 - b. If the mitigation strategy is unsuccessful, revise the mitigation strategy and perform Step 1 again.

At the end of this step, you have successfully tested all of your mitigation strategies in your test environment and can move to your pilot deployment environment.

2. Perform the following steps in the pilot deployments for each of the applications for which

Microsoft ACT: Phase3 - Testing and Mitigating Issues

you have developed mitigations:

- a. Test the mitigation strategy in your pilot deployment.
- b. If the mitigation strategy is unsuccessful, revise the mitigation strategy and perform Step 2 again.

At the end of this step, you have successfully tested all of your mitigation strategies in your pilot environment.

Reporting the Compatibility Mitigation Status to Stakeholders

After testing your application mitigation package, you must communicate your status to the appropriate stakeholders before deployment begins. We recommend that you perform this communication by using the following status ratings.

- **Resolved application compatibility issues.** This status indicates that the application compatibility issues are resolved and that these applications represent no risk to your environment.
- **Unresolved application compatibility issues.** This status indicates that there are unresolved issues for the specifically defined applications. Because these applications are a risk to your environment, more discussion is required before you can resolve the compatibility issues.
- **Changes to user experience.** This status indicates that the fix alters the user experience for the defined applications, possibly also requiring your staff to receive further training. More discussion is required before you can resolve the compatibility issues.
- **Changes in help desk procedures and processes.** This status indicates that the fix requires changes to your help desk procedures and processes, possibly also requiring your support staff to receive further training. More discussion is required before you can resolve the compatibility issues.

Resolving Outstanding Compatibility Issues

If you have any application compatibility issues that were not resolved by automated mitigation or by modifying the application, resolve them by using one of the following methods:

- Apply specific compatibility modes or run the program as an Administrator by using the Compatibility Administrator.

Note

For more information about using the Compatibility Administrator to apply compatibility fixes and compatibility modes, see the "Using the Compatibility Administrator" topics in the ACT Help.

- Run the application in a virtual environment.
Run the application in a version of Windows that is supported by the application in a virtual environment, such as one that Windows Virtual PC or Virtual Server 2005 provides.
- Resolve application compatibility by using non-Microsoft tools.

Microsoft ACT: Phase3 - Testing and Mitigating Issues

If the application was developed in an environment other than Microsoft Visual Studio®, you must use non-Microsoft debugging and analysis tools to help resolve the remaining application compatibility issues.

- Outsource the application compatibility mitigation.

If your developers have insufficient resources to resolve the application compatibility issues, outsource the mitigation effort to another organization.