

# Microsoft Partner Program

---

Guidelines for products that appropriately target Microsoft platforms and technologies

## Microsoft Platform Test for ISV Solutions Application Test Specification

**Microsoft**

Version 0.96  
Published: October 2004  
Updated: November 2007

Microsoft does not make any representation or warranty regarding specifications in this document or any product or item developed based on these specifications. Microsoft disclaims all express and implied warranties, including but not limited to the implied warranties or merchantability, fitness for a particular purpose and freedom from infringement. Without limiting the generality of the foregoing, Microsoft does not make any warranty of any kind that any item developed based on these specifications, or any portion of a specification, will not infringe any copyright, patent, trade secret or other intellectual property right of any person or entity in any country. It is your responsibility to seek licenses for such intellectual property rights where appropriate. Microsoft shall not be liable for any damages arising out of or in connection with the use of these specifications, including liability for lost profit, business interruption, or any other damages whatsoever. Some states do not allow the exclusion or limitation of liability or consequential or incidental damages; the above limitation may not apply to you. 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. This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS DOCUMENT.

Microsoft, Active Directory, MSDN, Win32, Windows, the Windows logo, Windows XP, and Windows Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

© 2003-2007 Microsoft Corporation. All rights reserved.

---

## Contents

<b>Contents</b> .....	3
<b>Welcome</b> .....	4
<b>Checklist for the Platform Test for ISV Solutions</b> .....	6
<b>Microsoft Windows Client</b> .....	8
<b>Microsoft Windows Server</b> .....	14
<b>Web Services and the .NET Framework</b> .....	20
<b>Microsoft Office</b> .....	21
<b>Microsoft SQL Server</b> .....	24
<b>Managed Code</b> .....	26

---

## Welcome

Welcome to *Microsoft Platform Test for ISV Solutions Application Test Specification*, describing the technical requirements of the Platform Test for ISV Solutions. This variation of the Platform Test is intended for software solutions that are typically sold to customers for installation and operation on customer systems (i.e. not hosted by the ISV).

The Platform Test for ISV Solutions is one of the product tests for software applications from Independent Software Vendors (ISVs) that meet the certification requirements for ISVs to obtain the ISV/Software Solutions Competency of the Microsoft Partner Program. Please note that ISVs must also comply with the Customer Reference Requirements to obtain for the ISV/Software Solutions Competency. Please visit [partners.microsoft.com](http://partners.microsoft.com) for more information on the ISV/Software Solutions Competency.

The Platform Test for ISV Solutions consists of two foundation test components, and four elective test components. The foundation test components of the Platform Test for ISV Solutions are the Windows Server and Windows Client test components. The Windows Server test component lists the requirements for server applications that run on Windows Server 2003. The Windows Client test component lists the requirements for desktop applications that run on Windows XP. The elective test components are Microsoft Office, SQL Server, Web Services plus .NET Framework, and Managed Code.

In order for a software application to pass the Platform Test for ISV Solutions, it needs to pass at least two test components, including at least one foundation test component. E.g. if a server application passes testing for the Windows Server component, then the application must also pass testing for any one of the SQL Server, Web Services plus .NET Framework, Microsoft Office, Managed Code, or Windows Client test components.

**Note:** Only client/server applications are eligible for testing for both the Windows Client and Windows Server test components of the Platform Test for ISV Solutions. A client/server application is defined as an application that includes code that loads into memory and runs on the Windows Client platform and usually different code that loads into memory and runs on the Windows Server platform, and there is interaction between the code running on the two platforms. Stand-alone applications that are supported on both the Windows Client and Windows Server platforms can only be tested (and earn Microsoft Partner Points) for one of the foundation test components.

The Platform Test for ISV Solutions helps ISVs earn Microsoft Partner Points, which help ISVs increase their status in the Microsoft Partner Program. The individual test components of the Platform Test for ISV Solutions are worth 10 Partner Points each, but only if the application passes testing for the Platform Test for ISV Solutions overall (i.e. passes one fundamental test component plus one other test component). For example, an application that passes the Windows Client, Managed Code, and MS Office test components, would earn 30 Partner Points, while an application that only passes the Windows Client test component would not earn any Partner Points.

**Note:** Microsoft restricts the number of points awarded for tested products. ISVs can earn Microsoft Partner Points for one version of one application that has passed

qualified tests (including the Platform Test for ISV Solutions). ISVs can increase the number of partner points earned by having that one version of one application pass additional tests (or additional test components of the Platform Test for ISV Solutions). However, ISVs do not accrue points by having multiple applications or multiple versions of one application pass testing. If an ISV submits multiple applications or multiple versions of one application for testing, Microsoft will only award points for the one version of the one application that has the most points associated with it.

Additionally, Microsoft caps the number of points awarded for tested products.

**Note:** ISVs cannot earn Microsoft Partner Points by having a product tested for the Platform Test for ISV Solutions, as well as the Platform Test for Hosted Solutions. ISVs only earn points for one of these tests, not both.

As mentioned previously, the Platform Test for ISV Solutions helps ISVs obtain the ISV/Software Solutions Competency of the Partner Program. Microsoft partners can earn additional Partner Points by joining competencies of the Microsoft Partner Program .

Please visit [partners.microsoft.com](https://partners.microsoft.com), or contact Microsoft (see <https://partners.microsoft.com/PartnerProgram/ContactMicrosoft.aspx>) for more information on Microsoft Partner Points.

### **Applications Eligible for the Platform Test for ISV Solutions**

Only released versions of commercially available software are eligible to be tested for the Platform Test for ISV Solutions. Beta versions of applications, custom-developed applications, and applications developed in-house for internal use are not eligible to be tested for the Platform Test for ISV Solutions.

Development tools that feature an integrated development environment (IDE) are eligible to be tested for the Platform Test for ISV Solutions. Applications that are add-ons, or plug-ins for development tools, that extend the IDE of the development tools they plug into are also eligible to be tested for the Platform Test for ISV Solutions. However, products that consist only of software libraries that are intended to be incorporated into new applications are not.

---

## Checklist for the Platform Test for ISV Solutions

### [Microsoft Windows Client](#)

#### **Fundamental Requirements**

Desktop applications must comply with all fundamental requirements to pass testing for this test component

1. Executes on Microsoft Windows XP and maintains stability while performing primary functionality
2. Uses Windows Resources (heaps, page heaps, locks, and handles) appropriately
3. Does not attempt to replace files protected by Windows File Protection
4. All device or filter drivers installed by the application are digitally signed by Microsoft WHQL
5. All kernel-mode drivers installed by the application pass Windows driver verification

#### **Elective Requirements**

Desktop applications must comply with any one of the elective requirements to pass testing for this test component

1. Does not require a reboot during installation, operation, or removal
2. Provides installation program that supports "All Users" installs
3. Remains stable while performing Fast User Switching
4. Supports use by a Limited User

**Note:** Installation of products should be intuitive and easy to follow by anyone with typical Administrator abilities. Products must include a documented installation procedure, preferably with an automated installation routine.

### [Microsoft Windows Server](#)

#### **Fundamental Requirements**

Server applications must comply with all fundamental requirements to pass testing for this test component

1. Executes on Microsoft Windows Server 2003 and maintains stability while performing primary functionality
2. Uses Windows Resources (heaps, page heaps, locks, and handles) appropriately
3. Does not attempt to replace files protected by Windows File Protection
4. All device or filter drivers installed by the application are digitally signed by Microsoft WHQL
5. All kernel-mode drivers installed by the application pass Windows driver verification

#### **Elective Requirements**

Server applications must comply with any one of the elective requirements to pass testing for this test component

1. Does not require a reboot during installation, operation, or removal
2. Does not disable other services during installation, operation, or removal
3. Supports Active Directory
4. Supports Windows Management Instrumentation (WMI)

5. Utilizes Windows SharePoint Services
6. Utilizes ASP.NET for Web Applications

**Note:** Installation of products should be intuitive and easy to follow by anyone with typical Administrator abilities. Products must include a documented installation procedure, preferably with an automated installation routine.

### Web Services and the .NET Framework

Applications must comply with any one requirement to pass testing for this test component

1. Exposes a Web service using .NET Framework
2. Consumes a Web service using .NET Framework

### Microsoft Office

Applications must comply with any one requirement to pass testing for this test component.

1. Application includes a **COM add-in** for Microsoft Office 2003 or 2007
2. Application includes a **VBA add-in** for Microsoft Office 2003 or 2007
3. Application includes a **VSTO add-in** for Microsoft Office 2003 or 2007
4. Application exposes data in Microsoft Office 2003 or 2007 via **Research and Reference feature**
5. Application integrates data in Microsoft Office 2003 or 2007 via **Smart Tags**
6. Application requires one of the Microsoft Office 2003 Editions or Microsoft Office 2007 Editions programs to exercise some its documented functionality

### Microsoft SQL Server

#### **Fundamental Requirements**

Applications must comply with all fundamental requirements to pass testing for this test component

1. Supports ADO, ADO.NET, OLE DB, ODBC, or JDBC to connect to SQL Server 2005
2. Supports SQL Server Authentication or Windows Authentication

### Managed Code

Applications must comply with all requirements to pass testing for this component

1. All Application Runtime Assemblies consist of Managed Code built on the Microsoft .NET Framework.

---

## Microsoft Windows Client

The Microsoft Windows Client component of the Microsoft Platform Test for ISV Solutions is intended to identify desktop applications that run on the Microsoft Windows Operating System. During this test, a typical installation of the application will be performed. The test bed will include either Microsoft Windows XP Professional, or Microsoft Windows XP Home Edition. All the latest critical security patches will be applied to the system prior to installing the application.

**Note:** Only client/server applications are eligible for testing for both the Windows Client and Windows Server test components of the Platform Test for ISV Solutions. A client/server application is defined as an application that includes code that loads into memory and runs on the Windows Client platform and usually different code that loads into memory and runs on the Windows Server platform, and there is interaction between the code running on the two platforms. Stand-alone applications that are supported on both the Windows Client and Windows Server platforms can only be tested (and earn Microsoft Partner Points) for one of the foundation test components.

### Fundamental Requirements

Applications must meet all the Fundamental Requirements.

#### 1. Execute on Microsoft Windows XP and maintain stability while performing primary functionality.

##### *Test Objectives*

Applications must execute on Microsoft Windows XP and perform their primary functions as expected without crashing or causing the user's computer to crash, fail or function improperly.

A crash is any failure within a server component or service that either causes data loss or forces unscheduled downtime of the server or service. A crash within a client component or utility component is considered to be an application failure that prevents the user from continuing. A failure within a server component or service will not be considered a crash if it meets both of the following conditions:

- a) does not cause loss of data,
- b) does not force shutdown or unscheduled downtime for any server or service.

A failure within a client component or tool will not be considered a crash if it meets all three of the following conditions:

- a) does not cause loss of data,
- b) displays information that would allow a typical user to understand what went wrong and how to avoid the problem in the future
- c) allows the user to continue running the application or close it.

#### 2. Use Windows resources (heaps, page heaps, locks, and handles) appropriately.

##### *Test Objectives*

The heap, critical sections, and handles can be misused, resulting in less reliable applications and failures with subtle circumstances that affect customers but may not be easily reproducible. You can easily test each of these items to ensure they are not misused. Applications must not misuse these resources in any way that could ever have potential negative consequences.

### *Heap Use*

Dynamic memory allocations come from the heap. Heap errors can result in security holes and can cause an application to fail. There are several invalid ways to use the heap, including:

- Allocating memory but writing beyond the end of the allocation (buffer overruns)
- Using allocated memory after it is freed
- Freeing an allocation twice
- Freeing unallocated memory
- Using wrong heap pointers

### *Critical Section Use (Locks Usage Checking)*

Critical sections are user mode synchronization primitives that guarantee exclusive access to application data in a multithreaded environment. Invalid uses of critical sections include:

- Releasing a critical section that the current thread does not own
- Terminating threads while they own critical sections
- Using a critical section before being initialized
- Leaking critical sections (for example, did not call `DeleteCriticalSection`)
- Double initialized critical sections

### *Handle Use*

Kernel handles—including handles to files, events, and so on—can also be misused in the following ways:

- Reusing a handle after being closed
- Using a handle for an operation that requires another handle type (you cannot read from an event)
- Using a random handle value
- Using a null handle or a pseudo-handle—for example, values returned by `GetCurrentProcess()`—when it is not permitted

To see why these kinds of errors can have bad consequences, consider the example of reusing a handle after it is closed. When a handle is closed, the system will reuse the value previously assigned. Assume that you have a file handle open and you close it, but you keep the value of the handle in some global variable. If some other part of the process opens a file handle for a totally different reason, perhaps even from external

code, the new handle might contain the same value. If you still hold the old value in a variable and continue to use it, you may write in the wrong file.

**Note:** VeriTest uses Microsoft's Application Verifier (AppVerifier) tool for this test. To read more about AppVerifier, please visit [msdn.microsoft.com](http://msdn.microsoft.com), and search for "AppVerifier".

### 3. Do not attempt to replace files under Windows File Protection

#### *Test Objectives*

- Perform the initial application installation without attempting to replace any files protected by Windows File Protection (WFP).
- Perform any just-in-time installations without attempting to replace any files protected by Windows File Protection.

The application must not attempt to replace any files that are protected by Windows File Protection (WFP). To ensure that the application does not invoke WFP, it should call `SfcIsFileProtected` when installing any file that it did not create. The Windows Installer service does this automatically.

Protected files include the following files that ship on the Windows XP product CD:

- Most .SYS, .DLL, .EXE and .OCX files.
- The following fonts: `Micross.ttf`, `Tahoma.ttf`, `Tahomabd.ttf`, `Dosapp.fon`, `Fixedsys.fon`, `Modern.fon`, `Script.fon`, and `Vgaoem.fon`.

**Note:** Some redistributable files, such as specific versions of Microsoft Foundation Classes (MFC) DLLs, are installed by Windows XP and are protected by WFP.

Protected files form the core of the operating system and it is essential for system stability that the proper versions be maintained. These files can only be updated through service packs, operating system upgrades, Quick Fix Engineering (QFE) hot-fixes, and Windows Update. Applications cannot replace them, and attempting to replace these files by any means other than those listed above will result in the files being restored by the Windows File Protection feature (see the subsection About Windows File Protection, below).

If the application requires newer versions of these components, it must update these components by using a Microsoft Service Pack that installs the required versions.

**EXAMPLE:** When Microsoft publishes an update to DirectX, it will be provided in a package (either a Windows service pack or its own service pack). An application including the updated DirectX must use the package install and not attempt to directly install files from the package. Installing individual files is not allowed; in addition, Windows File Protection would prevent it and the user experience would be poor.

#### *About Windows File Protection*

Windows File Protection is a feature of Windows XP that prevents the unauthorized replacement of essential system files. WFP runs as a background process on

Windows XP and monitors the files listed earlier in this section. When WFP detects that a protected file has been changed, it restores the original.

Do not prompt the user to update or delete any Windows File Protected components.

**Note:** Attempting to install components that are under Windows File Protection but have not yet been installed on the system will cause Windows File Protection to install the components. This is correct behavior.

#### **4. Any device or filter drivers that come with the application must pass the Windows Hardware Compatibility Test.**

##### *Test Objectives*

Any hardware device drivers or filter drivers for categories that the Windows Hardware Quality Labs (WHQL) accepts must pass the relevant tests in Windows Hardware Compatibility Test (HCT) 11.0 or later.

For certain categories of drivers, Windows XP warns end users if they attempt to install a driver that does not have a digital signature from Microsoft. Any drivers that the WHQL accepts must be digitally signed by Microsoft.

**Note:** For drivers that WHQL does not accept, the requirements in this section do not apply.

#### **5. Any kernel-mode drivers that the application installs must pass verification testing on Windows XP**

##### *Test Objectives*

Poorly written kernel-mode drivers have the potential to crash the system. Therefore, it is critical that any application that includes kernel-mode drivers, such as backup, copy protection and compact disc (CD) burning products, be thoroughly tested to minimize this risk.

#### **Elective Requirements**

Applications must meet one of the following Elective Requirements.

##### **1. Does not require a reboot during installation, operation, or removal**

###### *Test Objectives*

In Windows XP, very few installation situations require a reboot. Reboots are unwelcome by customers and, in some situations, can make deploying applications difficult. The application must not require or suggest a reboot during or after installation.

**Note:** Reboots required by a Windows approved Service Packs installed by the application are permitted. However, reboots required a GINA.DLL or certain filter drivers installed by the application are not permitted.

##### **2. Supports "All Users" Installs**

###### *Test Objectives*

Applications are often used by more than one user on the computer. To comply with this requirement, the application's installer must default to "all users" or provide an "all users" installation as an option. For example, an installer might default to the option of installing the application only for the current user but the application must provide an option to install for all users.

### 3. Supports Fast User Switching

#### *Test Objectives*

In Windows XP, the Fast User Switching feature allows multiple users sharing the same computer to have individual profiles and to swap their current work spaces without logging off. The application must not crash or lose data or settings when customers use Fast User Switching.

For example, if the first user has an editor application open and a subsequent user launches the same editor application, the first instance of the application must not shut down and must not lose any of the first user's edits.

If additional instances of the application run by separate users can result in failure of primary functionality, the application must do one of the following:

- Detect that it is already running under a separate user account and block the specific potentially problematic features, or
- Detect that it is already running and block all features of the application when launching subsequent instances of the application.

When blocking any feature to prevent failure under Fast User Switching, the application must inform the user why it did so.

### 4. Supports use by Limited User

#### *Test Objectives*

Applications must not require users to have unrestricted access (for example, Administrator privileges) to make changes to system or other files and settings. In other words, the application must function properly in a secure Windows environment (see below).

As long as a Limited User can successfully run the major features of the application, it is acceptable for minor features to fail gracefully. These minor features must not be installed by any default mechanism (for example, a minimal or typical install) other than a complete install and must not be considered important for the operation of the program. Examples of such minor features include components necessary to support legacy file formats.

A secure Windows environment is defined as the environment exposed to a Limited (non-Administrator) user by default on a clean-installed NTFS system. In this environment, users can only write to these specific locations on a local computer [Note 1]:

- Their own portions of the registry (HKEY\_CURRENT\_USER) [Note 2]

- Their own user profile directories (CSIDL\_PROFILE)
- A Shared Documents location (CSIDL\_COMMON\_DOCUMENTS) [Note 3]
- A folder that the user creates from the system drive root

However, applications defaulting to use of these folders do not comply with the other requirements of this section.

Users can also write to subkeys and subdirectories of these locations. For example, users can write to CSIDL\_PERSONAL (My Documents) because it is a subdirectory of CSIDL\_PROFILE. Users have read-only access to the rest of the system.

## NOTES

[1] Applications can modify the default security for an application-specific subdirectory of CSIDL\_COMMON\_APPDATA. This may provide an additional location to which users can write for a given application.

Any modification of the default security for an application-specific subdirectory of CSIDL\_COMMON\_APPDATA must be documented when submitting your application.

[2] Users cannot write to the following subsections of HKCU:

\Software\Policies

\Software\Microsoft\Windows\CurrentVersion\Policies

[3] By default, users cannot write to other users' shared documents; they can only read other users' shared documents. Applications can modify this default security on an application-specific subdirectory of CSIDL\_COMMON\_DOCUMENTS.

Any modification of the default security on an application-specific subdirectory of CSIDL\_COMMON\_DOCUMENTS must be documented when submitting your application.

---

## Microsoft Windows Server

The Microsoft Windows Server component of the Microsoft Platform Test for ISV Solutions is intended to identify server applications that run on the Microsoft Windows Server Operating System. During this test, a typical installation of the application will be performed. The test bed will include Microsoft Windows Server 2003. All the latest critical security patches will be applied to the system prior to installing the application.

**Note:** Only client/server applications are eligible for testing for both the Windows Client and Windows Server test components of the Platform Test for ISV Solutions. A client/server application is defined as an application that includes code that loads into memory and runs on the Windows Client platform and usually different code that loads into memory and runs on the Windows Server platform, and there is interaction between the code running on the two platforms. Stand-alone applications that are supported on both the Windows Client and Windows Server platforms can only be tested (and earn Microsoft Partner Points for) one of the foundation test components.

Stand-alone desktop applications that are installed in a shared folder on a file server, and run on Windows XP (e.g. via a mapped drive) cannot be tested for the Windows Server test component.

Desktop applications that connect to a database or write to/read from shared files on a file server cannot be tested for the Windows Server test component, unless the application includes server code that runs on Windows Server 2003 outside the context of the database. If the desktop application connects to SQL Server 2005, Microsoft Access 2003, or Microsoft Access 2007, then the application can be tested for the SQL Server or Microsoft Office test components, respectively.

### Fundamental Requirements

Applications must meet all the Fundamental Requirements.

#### 1. Execute on Microsoft Windows Server 2003 and maintain stability while performing primary functionality.

##### *Test Objectives*

Applications must execute on Microsoft Windows Server 2003 and perform their primary functions as expected without crashing or causing the user's computer to crash, fail or function improperly.

A crash is any failure within a server component or service that either causes data loss or forces unscheduled downtime of the server or service. A crash within a client component or utility component is considered to be an application failure that prevents the user from continuing. A failure within a server component or service will not be considered a crash if it meets both of the following conditions:

- a) does not cause loss of data,
- b) does not force shutdown or unscheduled downtime for any server or service.

A failure within a client component or tool will not be considered a crash if it meets all 3 of the following conditions:

- a) does not cause loss of data,

- b) displays information that would allow a typical user to understand what went wrong and how to avoid the problem in the future
- c) allows the user to continue running the application or close it.

## 2. Use Windows resources (heaps, page heaps, locks, and handles) appropriately.

### *Test Objectives*

The heap, critical sections, and handles can be misused, resulting in less reliable applications and failures with subtle circumstances that affect customers but may not be easily reproducible. You can easily test each of these items to ensure they are not misused. Applications must not misuse these resources in any way that could ever have potential negative consequences.

### *Heap Use*

Dynamic memory allocations come from the heap. Heap errors can result in security holes and can cause an application to fail. There are several invalid ways to use the heap, including:

- Allocating memory but writing beyond the end of the allocation (buffer overruns)
- Using allocated memory after it is freed
- Freeing an allocation twice
- Freeing unallocated memory
- Using wrong heap pointers

### *Critical Section Use (Locks Usage Checking)*

Critical sections are user mode synchronization primitives that guarantee exclusive access to application data in a multithreaded environment. Invalid uses of critical sections include:

- Releasing a critical section that the current thread does not own
- Terminating threads while they own critical sections
- Using a critical section before being initialized
- Leaking critical sections (for example, did not call `DeleteCriticalSection`)
- Double initialized critical sections

### *Handle Use*

Kernel handles—including handles to files, events, and so on—can also be misused in the following ways:

- Reusing a handle after being closed
- Using a handle for an operation that requires another handle type (you cannot read from an event)
- Using a random handle value
- Using a null handle or a pseudo-handle—for example, values returned by `GetCurrentProcess()`—when it is not permitted

To see why these kinds of errors can have bad consequences, consider the example of reusing a handle after it is closed. When a handle is closed, the system will reuse the value previously assigned. Assume that you have a file handle open and you close it, but you keep the value of the handle in some global variable. If some other part of the process opens a file handle for a totally different reason, perhaps even from external code, the new handle might contain the same value. If you still hold the old value in a variable and continue to use it, you may write in the wrong file.

**Note:** VeriTest uses Microsoft's Application Verifier (AppVerifier) tool for this test. To read more about AppVerifier, please visit [msdn.microsoft.com](http://msdn.microsoft.com), and search for "AppVerifier".

### 3. Do not attempt to replace files under Windows File Protection

#### *Test Objectives*

- Perform the initial application installation without attempting to replace any files protected by Windows File Protection (WFP).
- Perform any just-in-time installations without attempting to replace any files protected by Windows File Protection.

The application must not attempt to replace any files that are protected by Windows File Protection (WFP). To ensure that the application does not invoke WFP, it should call `SfcIsFileProtected` when installing any file that it did not create. The Windows Installer service does this automatically.

Protected files include the following files that ship on the Windows Server 2003 product CDs:

- Most .SYS, .DLL, .EXE and .OCX files.
- The following fonts: `Micross.ttf`, `Tahoma.ttf`, `Tahomabd.ttf`, `Dosapp.fon`, `Fixedsys.fon`, `Modern.fon`, `Script.fon`, and `Vgaoem.fon`.

**Note:** Some redistributable files, such as specific versions of Microsoft Foundation Classes (MFC) DLLs, are installed by Windows Server 2003 and are protected by WFP.

Protected files form the core of the operating system and it is essential for system stability that the proper versions be maintained. These files can only be updated through service packs, operating system upgrades, Quick Fix Engineering (QFE) hot-fixes, and Windows Update. Applications cannot replace them, and attempting to replace these files by any means other than those listed above will result in the files being restored by the Windows File Protection feature (see the subsection About Windows File Protection, below).

If the application requires newer versions of these components, it must update these components by using a Microsoft Service Pack that installs the required versions.

**EXAMPLE:** When Microsoft publishes an update to DirectX, it will be provided in a package (either a Windows service pack or its own service pack). An application including the updated DirectX must use the package install and not attempt to directly

install files from the package. Installing individual files is not allowed; in addition, Windows File Protection would prevent it and the user experience would be poor.

#### *About Windows File Protection*

Windows File Protection is a feature of Windows Server 2003 that prevents the unauthorized replacement of essential system files. WFP runs as a background process on Windows Server 2003 and monitors the files listed earlier in this section. When WFP detects that a protected file has been changed, it restores the original.

Do not prompt the user to update or delete any Windows File Protected components.

**Note:** Attempting to install components that are under Windows File Protection but have not yet been installed on the system will cause Windows File Protection to install the components. This is correct behavior.

#### **4. Any device or filter drivers that come with the application must pass the Windows Hardware Compatibility Test.**

##### *Test Objectives*

Any hardware device drivers or filter drivers for categories that the Windows Hardware Quality Labs (WHQL) accepts must pass the relevant tests in Windows Hardware Compatibility Test (HCT) 11.0 or later.

For certain categories of drivers, Windows Server 2003 warns end users if they attempt to install a driver that does not have a digital signature from Microsoft. Any drivers that the WHQL accepts must be digitally signed by Microsoft.

**Note:** For drivers that WHQL does not accept, the requirements in this section do not apply.

#### **5. Any kernel-mode drivers that the application installs must pass verification testing on Windows Server 2003**

##### *Test Objectives*

Poorly written kernel-mode drivers have the potential to crash the system. Therefore, it is critical that any application that includes kernel-mode drivers, such as backup, copy protection and compact disc (CD) burning products, be thoroughly tested to minimize this risk.

### **Elective Requirements**

Applications must meet one of the following Elective Requirements.

#### **1. Does not require a reboot during installation, operation, or removal**

##### *Test Objectives*

In Windows Server 2003 very few installation situations require a reboot. Reboots are unwelcome by customers and, in some situations, can make deploying applications difficult. The application must not require or suggest a reboot during or after installation.

**Note:** Reboots required by a Windows approved Service Packs installed by the application are permitted. However, reboots required a GINA.DLL or certain filter drivers installed by the application are not permitted.

## 2. Does not disable other services during installation, operation, or removal

### *Test Objectives*

An application and its installer must not cause services to become unavailable even temporarily, such as a service restart. Services are software components that the Service Control Manager (SCM) manages. They often provide resources to multiple applications and other components.

Unless an application (or its installer) inform the administrator and await guidance on scheduling the shutdown or reset, the only services an application may shut down or reset are services that are clearly part of the application, and are owned by the vendor. Services provided by third parties, other products, or the operating system should not be shut down casually.

In the context of this requirement, services might be hosted two ways:

- Within a process—for example, using W3svc within a worker process
- Outside a process—using SQL Server, Exchange, and so on

A service is considered unavailable when it:

- Is stopped in SCM and it is not for planned service downtime
- It does not respond within three client retries, or after a reasonable client time-out
- Any request to the service results in errors or no response or produces unexpected results

## 3. Supports Active Directory

### *Test Objectives*

Active Directory presents organizations with a directory service designed for distributed computing environments. Active Directory allows organizations to centrally manage and share information on network resources and users while acting as the central authority for network security. In addition to providing comprehensive directory services to a Windows environment, Active Directory is designed to be a consolidation point for isolating, migrating, centrally managing, and reducing the number of directories that companies require.

## 4. Supports Windows Management Instrumentation (WMI)

### *Test Objectives*

Windows® Management Instrumentation (WMI) is a component of the Microsoft Windows operating system and is the Microsoft implementation of Web-Based Enterprise Management (WBEM), which is an industry initiative to develop a standard technology for accessing management information in an enterprise environment. WMI uses the Common Information Model (CIM) industry standard to represent systems, applications, networks, devices, and other managed components.

You can use WMI to automate administrative tasks in an enterprise environment. WMI can be used in all Microsoft Windows-based applications, and is most useful in enterprise applications.

#### **5. Utilizes Windows® SharePoint™ Services**

Windows SharePoint Services allows teams to create Web sites for information sharing and document collaboration, benefits that help increase individual and team productivity. Windows SharePoint Services is a component of the Windows Server 2003 information worker infrastructure and provides team services and sites to Microsoft Office System and other desktop programs. It also serves as a platform for application development.

#### **6. Utilizes ASP.NET for Web Applications**

##### *Test Objectives*

ASP.NET is a programming framework built on the common language runtime that can be used on a server to build powerful Web applications.

---

## Web Services and the .NET Framework

Applications that utilize the Microsoft .NET Framework to expose or consume XML Web services that comply with industry Web service standards are eligible to be tested for this test component of the Platform Test for ISV Solutions. The WebService standards are currently defined as XML Schema 1.0, SOAP 1.1 and WSDL 1.1.

Applications must meet one of the following Elective Requirements.

### 1. Exposes an XML Web service using .NET Framework

#### *Test Objective*

Applications are required to fully support the following XML Web server standards when exposing programmable functionality:

- XML Schema 1.0
- SOAP 1.1
- WSDL 1.1
- UDDI 2.0 (required only if using a directory)

The exposed Web service must be written on the Microsoft .NET Framework.

### 2. Consumes a Web service using the .NET Framework

#### *Test Objective*

To comply with this requirement an application must use the .NET Framework to locate, reference, and use the functionality contained within a separate XML Web service. The client of an XML Web service is typically an application that is able to send, receive, and process messages to and from the XML Web service.

---

## Microsoft Office

The Microsoft Office component of the Microsoft Platform Test for ISV Solutions is intended to identify desktop applications that run on the Microsoft Windows Operating System that integrate with and/or leverage Microsoft Office. During this test, a typical installation of the application will be performed. The test bed will include Microsoft Office 2003 Editions or Microsoft Office 2007 Editions. Applications must require at least one of the programs included in Microsoft Office 2003 Editions or Microsoft Office 2007 Editions to exercise some of its documented functionality.

Applications must comply with any one requirement to pass testing for this component.

### 1. Application includes a COM add-in for Microsoft Office 2003 or 2007

#### *Test Objective*

A Component Object Model (COM) add-in is a dynamic-link library (DLL) that is specially registered for loading by the Microsoft Office applications. COM add-ins are built using any of the Office applications in Office Developer. In addition, you can create COM add-ins with Microsoft® Visual Basic® or Microsoft® Visual C++®. For more information about these tools, see the Microsoft Developer Network (MSDN®) Web site at <http://msdn.microsoft.com>.

COM add-ins use the Component Object Model that makes it possible to create a single add-in that is available to one or many of the Office applications. By developing COM add-ins, you can extend the functionality of your Office-based applications without adding complexity for users.

### 2. Application includes a VBA add-in for Microsoft Office 2003 or 2007

#### *Test Objective*

The suite of Microsoft Office applications incorporates the Visual Basic for Applications (VBA) programming environment to support automation to make them programmable. The VBA support for automation allows Office developers to use the features exposed through the object models of the entire Office suite of applications (as well as any third-party applications and components that support automation interfaces) as a set of business-application building blocks.

VBA and automation make it possible to integrate features from Office applications and other software components into a custom solution. VBA code running in one application can be used to create and work with objects from another installed application or component to create a sophisticated integrated solution. For example, VBA can be used to create an instance of Excel to use its mathematical or other functions code running in another application.

### 3. Application includes a VSTO add-in for Microsoft Office 2003 or 2007

#### *Test Objective*

Visual Studio Tools for Office (VSTO) extends .NET Framework development to Microsoft Office. VSTO allows developers to write code behind Word documents

and Excel workbooks with VB.NET or C# using the Visual Studio IDE. A VSTO project consists of two components: the Office document that acts as the “front-end” and the assembly (DLL) containing the compiled code from the project. The assembly is linked to the document by custom document properties. Keeping the code separate from the document facilitates deployment and maintenance.

When a document is opened the loader checks for custom properties. If these link to an assembly, and the Common Language Runtime (CLR) is started. If the document is trusted the assembly (the code) is downloaded. If the code is trusted, the code executes.

A typical VSTO add-in for Microsoft Office consists of the Office document, the assembly, along with any other assemblies that are referenced in the assembly, and .NET security policies.

#### **4. Application exposes data in Microsoft Office 2003 or 2007 via Research and Reference feature**

##### *Test Objective*

The Research and Reference feature in Microsoft Office allows users to locate and use the information they need without leaving the application in which they are working. Research and Reference provides an enhanced integrated Microsoft Internet Explorer-based search functionality from within Office applications. The Research and Reference feature is powerful and broad enough to be used in place of, or in addition to web-based research and reference sites.

Office 2003 and Office 2007 include numerous sources of Research and Reference "right out of the box", including Dictionary, Thesaurus, MSN® Search, and Microsoft Encarta® Encyclopedia, and a number of third party services. Research and Reference is also a platform for organizations to build their own research and reference services and for third party research providers to build subscription services.

#### **5. Application integrates data in Microsoft Office 2003 or 2007 via Smart Tags**

##### *Test Objective*

Smart tags are a feature in Microsoft Office applications that allows text to be labeled with contextual information while users type. Smart tags are extensible, so you can create your own recognizable strings, category labels, and customizable actions for those categories as well. Dynamic, highly-interactive smart tags can be developed using a Component Object Model (COM)-based application development system such as Microsoft Visual Basic® or Microsoft Visual C++®. More information on smart tag development can be found in the Smart Tag Software Development Kit (SDK).

#### **6. Application requires one of the Microsoft Office 2003 Editions or Microsoft Office 2007 Editions programs to exercise some its documented functionality**

##### *Test Objective*

Microsoft Office 2003 Editions and Microsoft Office 2007 Editions support extensive development capabilities that enable ISVs to build solutions incorporating programs from this platform. Applications that call at least one of the Microsoft Office 2003 Editions or Microsoft Office 2007 Editions programs listed below as some of its documented functionality is exercised comply with this requirement: Access, Excel, FrontPage, InfoPath, Live Meeting, OneNote, Outlook, PowerPoint, Project, Publisher, Small Business Accounting, Visio, Word.

Examples of types of applications that are eligible to be tested for this requirement include applications that use Access as their data repository, and applications that create reports in e.g. spreadsheets or documents, and call Excel or Word, respectively, to allow the user to view the report. Please note that applications that save data or reports as Excel spreadsheets or Word documents, but do not call these Office programs would not pass testing for the Microsoft Office test component.

---

## Microsoft SQL Server

The Microsoft SQL Server component of the Microsoft Platform Test for ISV Solutions is intended to identify applications that integrate with Microsoft SQL Server 2005.

**Note:** Applications cannot be tested for the SQL Server test component on Microsoft SQL Server 2005 Express Edition (SQL Server Express). Applications that normally install and work with this redistributable version of SQL Server will be reconfigured during the test to connect to SQL Server 2005.

**Note:** SQL Server 2005 ships with stronger username and password requirements than SQL Server 2000. If your application requires a specific user account, changes to the security schema that would weaken the default setting are not allowed.

For example, even though your application runs fine on SQL Server 2000 using a weak username and password combination, it may not install or run on SQL Server 2005 with the default security setting. It is not acceptable to alter the default settings of SQL Server 2005 to weaken the security requirements to allow your application to install and run properly. It is acceptable to alter the default security setting if the application requires a stronger security setting.

### Fundamental Requirements

Applications must meet all the Fundamental Requirements.

#### 1. Application supports ADO, ADO.NET, OLE DB, ODBC, or JDBC to connect to SQL Server

##### *Test Objectives*

Applications must use at least one of the supported methods to connect to SQL Server.

#### 2. Applications supports SQL Server Authentication or Windows Authentication

##### *Test Objectives*

##### SQL Server Authentication

When a user connects with a specified login name and password from a nontrusted connection, SQL Server performs the authentication itself by checking to see if a SQL Server login account has been set up and if the specified password matches the one previously recorded. If SQL Server does not have a login account set, authentication fails and the user receives an error message.

##### Windows Authentication

When a user connects through a Windows Server 2003 user account, SQL Server revalidates the account name and password by calling back to the Windows Operating System for the information.

SQL Server achieves login security integration with the Windows Operating System by using the security attributes of a network user to control login access. A user's

network security attributes are established at network login time and are validated by a Windows domain controller. When a network user tries to connect, SQL Server uses Windows-based facilities to determine the validated network user name. SQL Server then verifies that the person is who they say they are, and then permits or denies login access based on that network user name alone, without requiring a separate login name and password.

---

## Managed Code

*Managed Code* is code that targets the common language runtime of the .NET Framework. The common language runtime is the foundation of the .NET Framework. Code management is a fundamental principle of the runtime. The runtime manages code at execution time, providing core services such as memory management, thread management, and remoting, while also enforcing strict type safety and other forms of code accuracy that promote security and robustness. While code that targets the runtime is known as managed code, code that does not target the runtime is known as unmanaged code.

### Fundamental Requirements

Applications must meet all the Fundamental Requirements.

- 1. All application runtime assemblies (.EXE, .DLL, etc.) consist of Managed Code that is built on the Microsoft .NET Framework.**

**Note:** It is acceptable for assemblies sourced from third parties that are installed with the application under test to be unmanaged code.

**Note:** It is acceptable for applications to include non-runtime assemblies (e.g. data import utilities, or uninstaller routines) that are not managed code.