

Microsoft Partner Program

Guidelines for products that appropriately target Microsoft platforms and technologies

Microsoft Platform Test for ISV Solutions Application Test Specification

Microsoft

Version 0.51
Published: TBD
Updated: 12 May 2004

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 Microsoft Corporation. All rights reserved.

Contents

Contents	3
Welcome	4
Checklist for the Platform Test for ISV Solutions	5
Microsoft Windows Client	7
Microsoft Windows Server	13
Web Services and the .NET Framework	18
Microsoft Office	19
Microsoft SQL Server	21
Managed Code	22

Welcome

Welcome to *Microsoft Platform Test for ISV Solutions Application Test Specification*, describing the technical requirements for software applications for Independent Software Vendors (ISVs) to qualify for the ISV/Software Solution Competency of the Microsoft Partner Program.

The Platform Test consists of two foundation components, and four elective components. A software application is required to pass two components of the Platform Test, including at least one foundation component, for an ISV to comply with the Certification Requirements of the ISV/Software Solution Competency. ISVs must also comply with the Customer Reference Requirements to qualify for the ISV/Software Solution Competency. Please visit <http://members.microsoft.com/partner/program/competencies/isvsolutions.aspx> for more information on the ISV/Software Solution Competency.

ISVs earn 50 Microsoft Partner Points for enrolling in the ISV/Software Solution Competency. In addition, ISVs earn 10 Partner Points for each component their application passes. ISVs may choose to have their application tested for additional components of the Platform Test to earn more Microsoft Partner Points. Please visit <http://members.microsoft.com/partner/program/partnerpoints/default.aspx> for more information on Microsoft Partner Points.

The foundation components of the Platform Test are the Windows Server and Windows Client components. The Windows Server component lists the requirements for server applications that run on Windows Server 2003 or Windows 2000 Server. Please note that after 2004, Windows Server 2000 will no longer be an option for the Windows Server component. The Windows Client component lists the requirements for desktop applications that run on Windows XP. The elective components are Microsoft Office, SQL Server, Web Services plus .NET Framework (.NET Connected logo test), and Managed Code.

For an ISV to comply with the Certification Requirements for the ISV/Software Competency, its application must pass the Windows Server or the Windows Client Component, plus one additional 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 components.

This document lists the test requirements for each of the components of the Platform Test.

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 component

1. Executes on Microsoft Windows XP and maintains stability while performing primary functionality
2. Uses Windows Resources (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

Optional Requirements

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

1. Executes on Microsoft Windows Server 2003 or Windows 2000 Server and maintains stability while performing primary functionality
2. Uses Windows Resources (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

Optional Requirements

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

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

Microsoft Office

Applications must comply with any one requirement to pass testing for this component. Application requires at least one of the programs included in Microsoft Office 2003 Editions to exercise some of its documented functionality. Eligible applications must be one of the following:

1. Application includes a **COM add-in** for Microsoft Office 2003
2. Application includes a **VBA add-in** for Microsoft Office 2003
3. Application includes a **VSTO add-in** for Microsoft Office 2003
4. Application exposes data in Microsoft Office 2003 via **Research and Reference feature**
5. Application integrates data in Microsoft Office 2003 via **Smart Tags**

Microsoft SQL Server

Fundamental Requirements

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

1. Supports ADO, OLE DB, ODBC, or JDBC to connect to SQL Server

Optional Requirements

Applications must comply with any one of the optional requirements to pass testing for this component

1. Requires SQL Server 2000 SP3
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 Assemblies consist of Managed Code

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.

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, 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.

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.

Optional Requirements

Applications must meet one of the following Optional 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 Operating System. During this test, a typical installation of the application will be performed. The test bed will include either Microsoft Windows 2000 Server or Windows Server 2003. All the latest critical security patches will be applied to the system prior to installing the application.

Fundamental Requirements

Applications must meet all the Fundamental Requirements.

1. Execute on Microsoft Windows Server 2003 or Windows 2000 Server and maintain stability while performing primary functionality.

Test Objectives

Applications must execute on Microsoft Windows Server 2003 or Windows 2000 Server 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, 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.

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 or Windows 2000 Server 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 Windows 2000 Server 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 and Windows 2000 Server that prevents the unauthorized replacement of essential system files. WFP runs as a background process on Windows Server 2003 and Windows 2000 Server 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 and Windows 2000 Server 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 or Windows 2000 Server

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.

Optional Requirements

Applications must meet one of the following Optional Requirements.

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

Test Objectives

In Windows Server 2003 and Windows 2000 Server, 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

.NET Connected applications utilize the Microsoft .NET Framework or .NET Compact Framework to expose or consume XML Web services that comply with industry Web service standards. These standards are currently defined as XML Schema 1.0, SOAP 1.1 and WSDL 1.1.

Applications that pass the .NET Connected Component of the Platform Test are eligible to enroll in the Microsoft .NET Connected logo program. Further information regarding the .NET Connected logo program can be found at the following URL:

www.microsoft.com/net/logo.

Applications must meet one of the following Optional Requirements.

1. Exposes an XML Web service using .NET Framework (Fx) or .NET Compact 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 or .NET Compact 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. During this test, a typical installation of the application will be performed. The test bed will include Microsoft Office 2003 Editions. Applications must require at least one of the programs included in Microsoft Office 2003 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

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

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

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 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 includes 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 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).

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. Microsoft SQL Server 2000 is used in the test bed for this test.

Fundamental Requirements

Applications must meet all the Fundamental Requirements.

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

Test Objectives

Applications must use supported methods to connect to SQL Server

Optional Requirements

Applications must meet one of the following Optional Requirements.

1. Application requires SQL Server 2000 SP3 (or later)

Test Objectives

Applications must ensure SQL Server 2000 SP3 or a newer Service Pack is installed on the system in order to function. SP3 includes patches for known vulnerabilities reported by users or discovered through ongoing testing.

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 2000 or 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 assemblies (.EXE, .DLL, etc.) consist of Managed Code**