Systems Alliance

VPP-2: System Frameworks Specification

Revision 5.4

February 24, 2010

VPP-2 Revision History

This section is an overview of the revision history of the VPP-2 specification.

Revision 1.0, July 15, 1994

This edition reflects a non-technical revision for style and format issues.

Revision 1.1, August 17, 1994

This edition reflects edits to technical omissions and inconsistencies between VPP documents.

Revision 2.0, November 28, 1994

This edition reflects corrections to technical edits done to the Revision 1.0 document and changes discussed at the November 1-4 TWG meeting.

Revision 3.0, February 3, 1995

This edition adds framework revision management as discussed at the January 31 thru February 3, 1995 TWG meeting

Revision 4.0, January 29, 1996

This edition adds the WIN95, WINNT, SUN and HPUX frameworks with their G equivalents.

Revision 4.1, December 4, 1998

This edition incorporates the Windows 98 Operating System as part of the WIN95 Framework and removes references to the VPP-5 Specification. The information regarding contacting the Alliance was also updated.

Revision 4.2, December 17, 1999

This edition updates the WIN95, WINNT, SUN, and HPUX frameworks to match the latest revisions of the VISA and VXI*plug&play* instrument driver specifications. The WIN (Windows 3.1) Framework has not been updated. References to Hewlett-Packard VEE were changed to Agilent Technologies VEE.

Draft Revision 5.0, October 4, 2005

This edition adds the WIN64 framework for use by VISA. We are keeping the older operating systems and frameworks in this document because the definitions remain valid. This does not imply ongoing active IVI support for those frameworks.

Draft Revision 5.0, April 14, 2006

This edition removes the following frameworks: WIN, GWIN, WIN95, GWIN95, HPUX, GHPUX, SUN, and GSUN. The definitions of those frameworks in versions 4.2 and earlier of this document remain valid, and vendors may continue to claim compliance with those frameworks.

Revision 5.0, October 12, 2006

Approved at IVI Board of Directors meeting.

Draft Revision 5.1, October 25, 2006

Adds Windows Vista to the list of supported operating systems.

Revision 5.2, February 14, 2008

Updated the introduction to reflect the IVI Foundation organization changes. Replaced Notice with text used by IVI Foundation specifications. Add MatLab to ADE lists. Change HP VEE to Agilent VEE.

Revision 5.3, October 16, 2008

Updates Section 7 to reflect WIN64 with Vista 64. Eliminates all references to XP 64.

Revision 5.4, February 24, 2010

Adds Windows 7 to the list of supported operating systems.

NOTICE

VPP-2: *System Frameworks Specification* is authored by the IVI Foundation member companies. For a vendor membership roster list, please visit the IVI Foundation web site at `www.ivifoundation.org`.

The IVI Foundation wants to receive your comments on this specification. You can contact the Foundation through the web site at `www.ivifoundation.org`.

Warranty

The IVI Foundation and its member companies make no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The IVI Foundation and its member companies shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Trademarks

Product and company names listed are trademarks or trade names of their respective companies.

No investigation has been made of common-law trademark rights in any work.

Contents

Section 1
Introduction to the VXI*plug&play*
Systems Alliance and the IVI Foundation

The VXI*plug&play* Systems Alliance was founded by members who shared a common commitment to end-user success with open, multivendor VXI systems. The alliance accomplished major improvements in ease of use by endorsing and implementing common standards and practices in both hardware and software, beyond the scope of the VXIbus specifications. The alliance used both formal and de facto standards to define complete system frameworks. These standard frameworks gave end-users "plug & play" interoperability at both the hardware and system software level.

The IVI Foundation is an organization whose members share a common commitment to test system developer success through open, powerful, instrument control technology. The IVI Foundation's primary purpose is to develop and promote specifications for programming test instruments that simplify interchangeability, provide better performance, and reduce the cost of program development and maintenance.

In 2002, the VXI*plug&play* Systems Alliance voted to become part of the IVI Foundation. In 2003, the VXI*plug&play* Systems Alliance formally merged into the IVI Foundation. The IVI Foundation has assumed control of the VXI*plug&play* specifications, and all ongoing work will be accomplished as part of the IVI Foundation.

All references to VXI*plug&play* Systems Alliance within this document, except contact information, were maintained to preserve the context of the original document.

Section 2
Overview of System Frameworks

2.1 Introduction

This section introduces the *System Frameworks Specification*. It describes the intended audience and usage of the specification. It also gives references and other information, which although external to this specification, are critical to its understanding.

2.2 Objectives of the Specification

The *System Frameworks Specification* ensures that a VXI*plug&play* system can be assembled without concern for the compatibility or interoperability of the selected components. Each VXI*plug&play* system component conforms to one or more system frameworks. The system designers select the frameworks that meet their needs. They then select VXI*plug&play* components that conform to the selected frameworks. The requirements of the frameworks ensure the compatibility of the components.

The *System Frameworks Specification* sufficiently describes the system-level interfaces within each framework so that a vendor can ensure component compliance and compatibility. Each framework is broad enough to capture a significant number of users, ensuring its support by many vendors. However, if the framework is too broad, it places an unacceptable burden on the vendors, forcing them to supply support for environments that they deem unnecessary. The selection of the number and size of the system frameworks is a compromise between these needs.

2.3 Audience of the Specification

The primary audience for this specification consists of developers of system components—either component vendors, system integrators, or end users, who want to implement system components that are compliant with this specification. This specification may also be of interest to end users who want to know the detailed requirements of a particular system framework. However, the intent of the *System Frameworks Specification* is to ensure end-user success without any knowledge of this specification. Each framework is uniquely identified by a framework name, so that the system designer or end user can select components that conform to a selected VXI*plug&play* system framework.

2.4 Scope and Organization of the Specification

This specification is organized in sections. Each section discusses a particular independent level of implementation. The first sections give background and common framework information, and subsequent sections describe the details of the framework components, interfaces, and individual frameworks.

Section 1 explains the VXI*plug&play* Systems Alliance and its relation to the IVI Foundation.

Section 2 summarizes this specification and discusses its objectives, scope and organization, application, references, definition of terms and acronyms, and conventions.

Section 3 gives an introduction to system frameworks and includes the general definition of a system framework, why frameworks are necessary, and how frameworks are defined.

Section 4 defines common framework components, which are included in all defined VXIplug&play system frameworks.

Section 5 defines the frameworks which support components which are neither controllers nor instruments.Sections 6 and above define specific frameworks.

2.5 Assumptions

This specification assumes some familiarity with and understanding of VXI modules, the PC and UNIX operating systems, the VISA specifications (VPP-4.x), the *Instrument Driver Specifications* (VPP-3.x), and selected application development environment software applications including LabWindows, LabWindows/CVI, LabVIEW, Agilent VEE, MATLAB, Microsoft and Borland C, Visual Basic languages, and others.

2.6 Application of the Specification

This specification is intended for developers of VXI*plug&play* system components. It contains definitions and descriptions of all the components required to develop a complete VXI system. For each defined framework, explicit rules should be followed for the vendor to claim compliance to the framework. These rules ensure end-user success when that framework is selected for a user application.

2.7 References

Other documents were used as source material in the creation of this specification. In addition, other VXI*plug&play* documents may be of interest as you read this specification. These other related documents are as follows:

- VPP-1   *Charter Document*

- VPP-3.x   *Instrument Driver Specifications*

- VPP-4.x   *VISA Virtual Instrument Software Architecture Specifications*

- VPP-6   *Installation and Packaging Specification*

- VPP-7   *Soft Front Panel Specification*

- VPP-9   *Instrument Vendor Abbreviations*

- IEEE 1155   VXIbus Specification

2.8 Definitions of Terms and Acronyms

The following are some commonly used terms within this document.

- ADE   Application Development Environment

- GPIB   General Purpose Interface Bus (IEEE 488)

- VXI   VMEbus Extensions for Instrumentation (IEEE 1155)

- VISA   Virtual Instrument Software Architecture

- MXI   Multisystem Extension Interface

2.9 Conventions

The following headings appear on paragraphs throughout this specification. These headings give special meaning to the paragraphs.

Rules must be followed to ensure compatibility with the system framework. A rule is characterized by the words **SHALL** or **SHALL NOT** in bold upper case characters. These words are not used in this manner for any other purpose.

Recommendations contain advice to implementers. This advice affects the usability of the final device. Recommendations are included in this standard to draw attention to particular characteristics that the authors believe to be important to end-user success.

Permissions authorize specific implementations or uses of system components. A permission is characterized by the word **MAY** in bold upper case characters. These permissions are granted to ensure that specific system framework components are well defined and can be tested for compatibility and interoperability.

Section 3
Introduction to System Frameworks

3.1 Introduction

This section gives a detailed description of the VXI*plug&play System Frameworks Specification*. It describes the need for standard frameworks, standard framework implementation philosophy, the organization of the frameworks structure, and the relationships of these frameworks to the other elements of VXI*plug&play*. The specific implementation of the *System Frameworks Specification* is described in the following paragraphs.

3.2 The VXI Standard

The VXI standard supports the integration of instrumentation from many vendors into a single mainframe. A primary goal of the VXI standard is interoperability between instruments. VXI instruments must meet well defined interface specifications to ensure system compatibility.

VXI provides many mechanisms that support instrument interoperability. It provides a standard backplane electrical interface, standard mechanical packaging, standard system management and communications methods, and limits on local electromagnetic radiation and susceptibility. All of these requirements guarantee hardware compatibility and interoperability.

3.3 Need for Further Standards

Although the VXI standard ensured hardware interoperability, it did not address the implementation of system software and other system interfaces. These interfaces connect hardware and software products to form an integrated system. Because of the broad array of computers, operating systems and programming languages, selecting components that work together to control a VXI system can be challenging.

A VXI system is unique in that individual instruments do not provide integral user interfaces such as knobs, buttons, or displays. System software must be written to perform the simplest of tasks within a VXI system. When two system components do not work together, identifying and resolving the problem can be tedious and difficult for the user.

```
+-------------------------------------+
|      Application Development         |
|           Environment               |
+-------------------------------------+      Instrument Drive
       ========================
+-------------------------------------+
|         Instrument Drivers          |      Communication I
+-------------------------------------+
       ========================
+-------------------------------------+
|         Communication Mgmt          |      Instrument I/F
+-------------------------------------+
    ====       ====       ====
+----------+----------+----------+
|   VXI    |   GPIB   |  OTHER   |
+----------+----------+----------+
```
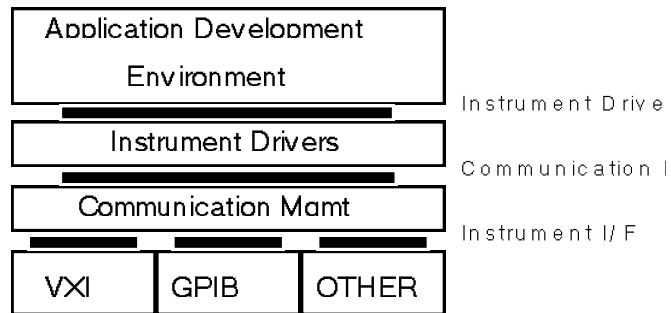
Figure 3-1 System Communication Interface

At least three software interfaces must work together to provide communications in a VXI system. These interfaces are the *instrument interface*, the *communications interface,* and the *instrument driver interface* as shown in Figure 3-1 System Communication Interface. Examples of the instrument interface are VXI Word Serial Protocol, VXI Register Based, GPIB, and RS-232. Each of these interfaces defines a unique protocol to transfer information between the instrument and the controller. The communications interface provides connection to each instrument interface in a standard way. Examples of this type of interface are NI-VXI/NI-488, EpConnect, and SICL. The instrument driver interface provides a link between a communication interface and a particular application program. The instrument driver adds additional abstraction to the instrument command language to reduce the knowledge and effort required to develop test applications.

All of the software modules and interfaces described above are dependent on the computer and operating system for which they have been designed. Even within the PC environment there are both DOS users and MS Windows users. Without additional standards, the system designer is required to determine the compatibility of all of the components outlined above. Because this is a difficult task, the designer often selects components from a single vendor with the hope that all components from that vendor are compatible. If compatibility problems arise, the designers know which vendor to call for help. Unfortunately, this single-vendor approach locks the designer into a closed system. It removes the benefits of the open, multi-vendor VXI systems approach.

3.4 System Frameworks

To address the problem of selection and integration of system components, VXIplug&play defines a number of system frameworks. Each framework supports a popular test system design approach. It has a specific name which can be referenced by system designers and advertised by component manufacturers. System designers can select a specific framework for their application and select from the available system components that conform to that framework.

Each system framework contains a number of components. Some of these components are included because they provide basic capabilities which other framework components depend on. Other VXIplug&play components are selected by the system designer or end user in a menu fashion, such as the application development environment.

A VXIplug&play system framework is a well-defined set of components. This set contains all of the necessary components to build a complete test system. The framework definition contains rules, recommendations, and permissions, all of which define the required compatibility and interoperability of each component in the set.

Specifically, each framework contains, but is not limited, to the following components:

- Requirements for the Control Computer Hardware

- Operating System

- VISA Interface and I/O Software

- Instrument Drivers

- Compatible Application Development Environments

- Required Documentation and Installation Support

- Soft Front Panel

- VXI Instruments, VXI Slot 0, System Controller, VXI Mainframe

Designing a system using a VXI*plug&play* system framework means that the selected components are compatible and interoperable. Following the framework requirements also ensures that all necessary system components have been included, resulting in a complete operational system.

3.5 Framework Definition

This document describes several system frameworks. The frameworks defined in this document are popular with VXI system designers and end users. As computer and software technology evolves, other hardware and software products may become popular within the VXI user community. As this change occurs, new frameworks will be defined to incorporate new capabilities. The next section, Common Framework Components, describes the requirements for the definition of new frameworks.

Section 4
Common Framework Components

4.1 Introduction

This section describes the system components and technologies that provide a foundation for all of the defined system frameworks. Where appropriate, it references other documents and standards which specifically define these components.

4.2 Instrument Interface

VXI, GPIB, RS-232, and other standards define protocols and methods for communicating information between instruments and a computer, which controls the actions of a test system. These protocols and methods are defined within their respective standards and are maintained by various organizations. The VXI*plug&play System Frameworks Specification* does not define any new communications standards but does require compatibility with specific standards for compliance.

RULE 4.1

> All VXI*plug&play* system frameworks **SHALL** support VXIbus instruments.

RECOMMENDATION 4.1

> VXI*plug&play* system frameworks should support IEEE 488 instruments.

4.3 Communication Interface

The communication interface links the instruments and the instrument drivers. Because of the central role of the communication interface, creating and implementing interface standards is critical to the success of VXI*plug&play*.

RULE 4.2

> All VXI*plug&play* systems frameworks **SHALL** require the VISA communication interface be utilized for VXI communication.

RULE 4.3

>   If a VXI*plug&play* system framework includes IEEE 488, then it **SHALL** require
>   the VISA communication interface be utilized for IEEE 488 communication.

OBSERVATION 4.1

To expedite the development and validation of the VXI*plug&play* VISA technology, the
VXI*plug&play* Alliance defined a series of software releases, each with expanding
capability. These initial releases were called the VISA Transition Library (VTL) and
provided a subset of the VISA capabilities. VTL maintained forward compatibility with
each release leading up to the release of VISA 1.0. VISA provides a complete
communication interface for VXIplug&play systems. Future releases of VISA will
enhance these capabilities and performance while maintaining forward binary
compatibility. As the VXI*plug&play* Alliance produces the next VISA release, the
Frameworks standard will be revised to require the currently released version of VISA.

RECOMMENDATION 4.2

>   Suppliers of the VISA communications interface software should provide
>   compatibility interfaces to as many of the existing interfaces as possible. These
>   include, but are not necessarily limited to, NI-VXI, EpConnect, and SICL.

4.4 Installation Media

To reduce the time and knowledge required to integrate a VXI system, VXIplug&play
defines a standard installation media patterned after industry standard implementations as
they relate to the framework.

The installation medium is shipped with each VXI*plug&play* product. It may also be
made available via the internet or other distribution mechanisms. It is responsible for the
correct installation of the instrument soft front panel, the instrument driver and other
VXI*plug&play* required software. It may also be responsible for other software
management tasks.

RULE 4.4

>   All VXI*plug&play* system frameworks **SHALL** include the requirement for a
>   VXI*plug&play* installation media as specified in VPP-6.

RULE 4.5

>   VXI*plug&play* Installation media SHALL be delivered with the VXI module.

4.5 Instrument Driver

An instrument driver links the communication interface and an application development environment. It provides a higher level, more abstract view of the instrument. It may also provide ADE-specific information that supports the capabilities of the ADE, such as a graphical representation.

RULE 4.6

> All VXI*plug&play* system frameworks **SHALL** include and define VXI*plug&play* instrument drivers.

4.6 Soft Front Panel

A VXIplug&play soft front panel is a graphical user interface for an instrument. It is used to verify instrument communications and functionality when the instrument is first integrated into the system. It may also be used as a learning tool to teach instrument control and capability concepts. The soft front panel may include additional capabilities that are not described here.

RULE 4.7

> All VXI*plug&play* system frameworks **SHALL** include and define VXI*plug&play* soft front panels.

4.7 Documentation

One of the goals of VXIplug&play is to reduce the time and complexity of integrating a test system based on VXI. To integrate an instrument into a VXI system, the instrument must be correctly configured. Instrument configuration information is typically included in a user manual. However, the organization and content of instrument manuals varies considerably from one manufacturer to the next.

RULE 4.8

> All VXI*plug&play* system frameworks SHALL include the standard VXI*plug&play* help file.

4.8 System Framework Support

To support the intent of the VXIplug&play charter, a module must support one or more system frameworks before it is considered to conform to VXI*plug&play*.

RULE 4.9

If a product does not meet all specified rules within a particular VXI*plug&play* framework, it **SHALL NOT** claim conformance to that framework.

RULE 4.10

VXI*plug&play* products **SHALL** conform to one or more VXI*plug&play* system frameworks.

RULE 4.11

VXI*plug&play* frameworks **SHALL** provide a revision number, which is updated when the framework is modified.

4.9 Framework Organization

Figure 4-1 represents the organization of the VXIplug&play system frameworks. They are separated by hardware platform, operating system, and programming style. The frameworks are named for the operating systems which they support. Each of the frameworks may be modified with the 'G' prefix to support the native G language of LabVIEW. For example the WINNT framework which directly supports the G language is called the GWINNT framework.

ANSI-C	MS C++ Borland C++ LabWindows/CVI	MS C++ Borland C++ LabWindows/CVI
Run-time link	MS C++ Borland C++ MS VB LabWindows/CVI LabVIEW Agilent VEE MATLAB	MS C++/C# Borland C++ MS VB.NET LabWindows/CVI LabVIEW Agilent VEE MATLAB
G	LabVIEW	LabVIEW
	WINNT	**WIN64**

Figure 4-1. VXI*plug&play* System Framework Organization

Section 5
VXI System Framework

5.1 Introduction

This section defines the requirements for VXIplug&play components which transcend all system frameworks. These components are utilized within a VXI*plug&play* system but do not place any additional requirements on other system components beyond the VXI specification.

5.2 VXI Mainframes

VXI mainframes are completely defined by the VXI Specification. They are utilized by all VXI*plug&play* frameworks and systems. The mainframe provides an environment for instrument modules, VXI slot 0 and interface modules. Because the mainframes provide system resources which are described by the knowledge base, a VXI*plug&play* mainframe must provide installation media. This information can be used by applications to automate the calculation of VXI power and cooling requirements as well as system validation.

RULE 5.1

> If a mainframe claims conformance to VXI*plug&play*, it **SHALL** identify which frameworks it supports. It **SHALL** provide VXI*plug&play* installation media in the format defined for each framework.

5.3 VXI Slot 0/Resource Manager

VXI Slot 0 modules provide required hardware resources for the operation of the VXIbus. They may provide translation between external communication interfaces and the VXI bus and protocols. VXI Slot 0 modules may also provide instrument functions which access VXI instrument busses.

RULE 5.2

> VXI Slot 0/Resource manager modules which provide a command set or equivalent **SHALL** identify which frameworks it supports and meet all requirements defined for that framework.

5.4 VXIplug&play Application Programs

VXIplug&play application programs are software applications composed of one or more executable files which are designed to be compatible with a particular VXI*plug&play* framework. VXI*plug&play* soft front panels are an example of an VXI*plug&play* application program.

RULE 5.3

> Application programs which claim conformance to VXI*plug&play* **SHALL** identify which frameworks it supports. It **SHALL** execute within the identified framework and utilize the VISA API exclusively for VXI communication.

RULE 5.4

> Application programs which claim conformance to VXI*plug&play* SHALL provide the application files on VXI*plug&play* style installation media appropriate for the framework identified.

Section 6
WINNT System Framework

6.1 Introduction

This section defines the specific requirements for the WINNT system framework, which includes the Windows 2000, Windows XP, Windows Vista 32, Windows Vista 64 (32-bit applications only), and Windows 7 (32-bit applications only) operating systems. It defines all of the unique components which are required to exist to support this framework. It also describes the optional components.

The WINNT framework supports 32 bit applications.

6.2 Overview of the Framework

The WINNT system framework defines a system based on the popular 32-bit Windows personal computer architecture and its compatibles. It is based on the Windows NT x86 operating system from Microsoft and supports the MS Windows graphical user interface. It uses the Visual Basic and C languages, application development environments as well as the 32 bit DLL technologies that support them.

6.3 Framework Revision

Current VPP specifications are used to comply with a VXI*plug&play* Framework.

OBSERVATION 6.1

Framework revisions were defined in earlier versions of VPP 2. Framework revisions were removed because the table of supporting specifications was not maintainable. Therefore, RULE 6.1 requiring specific revisions of supporting specifications was removed in revision 5.2 of this specification.

6.4 System Computer

The system computer includes the hardware support for execution of the application development environment, the instrument drivers, the communications interface and the users application. It may be embedded in the VXI mainframe, or it may be an external stand-alone PC.

A computer which supports VXI*plug&play* must support communications with VXI modules. A stand-alone PC may accomplish this by including appropriate PC/GPIB and GPIB/VXI bus interfaces, including a MXI bus interface between the PC and the VXI mainframe or other means of hardware translation. An embedded PC typically incorporates the necessary hardware interface as an integral part of its design.

RULE 6.2

> If a computer claims conformance to the VXI*plug&play* WINNT system framework, it **SHALL** be 100% 32-bit Windows compatible.

Rather than calling out specific system requirements (for storage, display, and memory) for this framework, this specification defers to the minimum operating system requirements as defined by Microsoft.

PERMISSION 6.1

> Vendors **MAY** add their own restrictions to further minimize the operating system requirements or service packs necessary for successful use of their instrument drivers.

RULE 6.3

> If a computer claims conformance to the VXI*plug&play* WINNT system framework, it **SHALL** have the capability to control VXI message-based and register-based instruments.

RULE 6.4

> If a computer claims conformance to the VXI*plug&play* WINNT system framework, it **SHALL** provide the VISA API as a 32-bit MS Windows compatible DLL named VISA32.DLL.

RECOMMENDATION 6.1

> VXI*plug&play* computers should have the capability to control GPIB instruments.

6.5 Operating System

The Windows 2000, Windows XP, Windows Vista 32, Windows Vista 64, or Windows 7 operating system is required for the languages supported in this framework.

RULE 6.5

> If a computer claims conformance to the VXI*plug&play* WINNT system framework, it **SHALL** provide Windows 2000, Windows XP, Windows Vista 32, Windows Vista 64, or Windows 7.

6.6 Instrument Drivers

The instrument drivers provided for this framework must support several popular application development environments. The common basis for these drivers is ANSI C.

RULE 6.6

> If a VXI module claims conformance to the VXI*plug&play* WINNT system framework,
> it **SHALL** provide ANSI C source code (`.c, .h files`),
> it **SHALL** provide the MS Windows 32 bit DLL library (`_32.dll, .def` files),
> it **SHALL** provide the Microsoft 32 bit DLL import library (`.lib` file),
> it **SHALL** provide the function panel file (`.fp file`),
> it **SHALL** provide the MS Visual Basic Function Declaration text file (`.bas` file), and
> it **SHALL** provide the driver documentation in a widely accepted documentation file format, such as Portable Document Format (`.pdf`) or compiled HTML (`.chm`).

6.7 Soft Front Panel

A *VXIplug&play* soft front panel provides a graphical user interface for an instrument. It is used to verify instrument communications and functionality when the instrument is first integrated into the system. It may also be used as a learning tool to teach instrument control and capability concepts. The soft front panel may include additional capabilities that are not described here. A VXI*plug&play* soft front panel is a VXI*plug&play* application which must only utilize the resources provided within the specified framework.

RULE 6.7

> If a VXI module claims conformance to the VXI*plug&play* WINNT system framework, it **SHALL** provide a 32-bit Windows stand-alone application that implements the VXI*plug&play* soft front panel.

6.8 Application Development Environments

The WINNT system framework supports, but is not limited to, the following languages with the revision described or higher revisions.

Company	Product	Revision
Borland	Turbo C/C++	4.5
Agilent Technologies	Agilent VEE	3.2
Microsoft	Visual Basic	6.0
The MathWorks	MATLAB	7.0
Microsoft	Visual C/C++	6.0
National Instruments	LabVIEW	7.0
National Instruments	LabWindows/CVI	4.0

PERMISSION 6.2

> Software products which claim 100% compatibility with the Microsoft Windows 32-bit DLL interface **MAY** claim compatibility with the WINNT system framework.

6.9 Installation Media

To reduce the time and knowledge required to integrate a VXI system, VXI*plug&play* defines standard installation media patterned after industry standard implementations such as MS Windows install programs.

RULE 6.8

The WINNT system framework installation media **SHALL** be compatible with the operating systems specified in the WINNT framework.

RULE 6.9

The WINNT system framework installation media **SHALL** be downloadable from the Internet or delivered on either a CD-ROM or a DVD. Other distribution media or mechanisms SHALL comply with VPP-6.

RULE 6.10

The WINNT system framework installation media **SHALL** be delivered with the VXI module.

Section 7
WIN64 System Framework

7.1 Introduction

This section defines the specific requirements for the WIN64 system framework, which includes the 64-bit editions of the Windows Vista and Windows 7 operating systems. It defines all of the unique components which are required to exist to support this framework. It also describes the optional components.

The WIN64 framework supports 64 bit applications. These applications may not be combined with WINNT framework (32 bit) applications.

7.2 Overview of the Framework

The WIN64 system framework defines a system based on the popular IBM personal computer architecture and its compatibles. It is based on several editions of the Windows Vista 64 and Windows 7 operating systems from Microsoft and supports the MS Windows graphical user interface. It uses the C language, application development environments as well as the 64 bit DLL technologies that support them.

7.3 Framework Revision

Current VPP specifications are used to comply with a VXI*plug&play* Framework.

OBSERVATION 7.1

> Framework revisions were defined in earlier versions of VPP 2. Framework revisions were removed because the table of supporting specifications was not maintainable. Therefore, RULE 7.1 requiring specific revisions of supporting specifications was removed in revision 5.2 of this specification.

7.4 System Computer

The system computer includes the hardware support for execution of the application development environment, the instrument drivers, the communications interface and the users application.

RULE 7.1

> If a computer claims conformance to the VXI*plug&play* WIN64 system framework, it **SHALL** be 100% Windows Vista 64 or Windows 7 (64-bit) compatible.

Rather than calling out specific system requirements (for storage, display, and memory) for this framework, this specification defers to the minimum operating system requirements as defined by Microsoft.

PERMISSION 7.1

> Vendors **MAY** add their own restrictions to further minimize the operating system requirements or service packs necessary for successful use of their instrument drivers.

RULE 7.2

> If a computer claims conformance to the VXI*plug&play* WIN64 system framework, it **SHALL** provide the VISA API as a DLL named visa64.dll.

RECOMMENDATION 7.1

> VXI*plug&play* computers should have the capability to control GPIB instruments.

7.5 Operating System

The Windows Vista 64 or Windows 7 (64-bit) operating system is required for the languages supported in this framework.

RULE 7.3

> If a computer claims conformance to the VXI*plug&play* WIN64 system
> framework, it **SHALL** provide a 64-bit edition of Windows Vista or Windows 7.

7.6 Instrument Drivers

The instrument drivers provided for this WIN64 framework must support several popular
application development environments. The common basis for these drivers is ANSI C.

RULE 7.4

If a VXI module claims conformance to the VXI*plug&play* WIN64 system framework,
it **SHALL** provide ANSI C source code (`.c`, `.h` files),
it **SHALL** provide the MS Windows 64-bit DLL library (`64.dll`, `.def` files),
it **SHALL** provide the Microsoft 64-bit DLL import library (`.lib` file),
it **SHALL** provide the function panel file (`.fp` file), and
it **SHALL** provide the driver documentation in a widely accepted documentation file
format, such as Portable Document Format (`.pdf`) or compiled HTML (`.chm`).

7.7 Soft Front Panel

A *VXIplug&play* soft front panel provides a graphical user interface for an instrument. It
is used to verify instrument communications and functionality when the instrument is first
integrated into the system. It may also be used as a learning tool to teach instrument
control and capability concepts. The soft front panel may include additional capabilities
not described here. A VXI*plug&play* soft front panel is a VXI*plug&play* application that
must utilize only the resources provided within the specified framework.

PERMISSION 7.2

> If a VXI module claims conformance to the VXI*plug&play* WIN64 system
> framework, it **MAY** provide a 64-bit Windows standalone application that
> implements the VXI*plug&play* soft front panel.

OBSERVATION 7.2

> It is not necessary to provide the 64-bit soft front panel, because the 32-bit soft
> front panel is already a requirement.

7.8 Application Development Environments

The WIN64 system framework supports, but is not limited to, the following languages with the revision described or higher revisions.

Company	Product	Revision
Microsoft	Visual Studio	2005/8.0
Agilent Technologies	Agilent VEE	*
National Instruments	LabVIEW	*
National Instruments	LabWindows/CVI	*
The MathWorks	MATLAB	*

* - The intent is to support these ADEs when a version compatible with this specification is released. These versions will be added as an editorial change.

PERMISSION 7.3

> Software products that claim 100 percent compatibility with the Microsoft Windows 64-bit DLL interface **MAY** claim compatibility with the WIN64 system framework.

7.9 Installation Media

To reduce the time and knowledge required to integrate a VXI system, VXI*plug&play* defines standard installation media patterned after industry standard implementations such as MS Windows install programs.

RULE 7.5

> The WIN64 system framework installation media **SHALL** be compatible with the operating systems specified in the WIN64 framework.

RULE 7.6

> The WIN64 system framework installation media **SHALL** be downloadable from the Internet or delivered on either a CD-ROM or a DVD. Other distribution media or mechanisms SHALL comply with VPP-6.

RULE 7.7

The WIN64 system framework installation media **SHALL** be delivered with the VXI module.

Section 8
GWINNT, GWIN64

8.1 Introduction

The G language provided by LabVIEW is supported by the GWINNT and GWIN64 system frameworks. This section defines the specific requirements for the application of the G language to these frameworks and the framework designations. It defines the unique components which are required to exist to support G within the framework.

8.2 Using the Bindings

Support for the G language within the VXI*plug&play* frameworks requires changes to the instrument driver files. The ANSI-C instrument drivers are replaced with the equivalent native LabVIEW .llb drivers as defined by the VPP-3.x specifications. The required help file is replaced with a help file for the LabVIEW driver.

8.3 Identifying a G Bound Framework

When the G binding is applied to a framework, the name of the framework is changed to reflect its different capabilities. A G prefix is added to the native framework name. WINNT becomes GWINNT, etc.

RULE 8.1

> When the G binding is applied to a base framework, the framework designation **SHALL** be the base framework name with the upper case letter G prefixed to it.

8.4 Instrument Drivers for a G Bound Framework

When the G binding is applied to a framework, the instrument driver must be supplied as a LabVIEW G language driver. The LabVIEW driver replaces the ANSI-C driver of the base framework. A framework specific help file is still required. It describes the LabVIEW driver.

RULE 8.2

When the G binding is applied to a base framework, the instrument driver files **SHALL**
be replaced with a LabVIEW instrument driver as specified in VPP-3.2.