Understanding Web Technologies in Measurement Applications Using VEE 5.0

Abstract

This white paper will explain how engineers can use the World Wide Web (WWW) to enable easy access to information pertinent to their jobs. It will briefly explain the more commonly used terms and technologies of the web and how HP VEE 5.0 and other programs can use these technologies to enable the easy dissemination of information to interested parties. This paper presumes a prior knowledge of HP VEE.

So just what ***is*** the Web? The World Wide Web Consortium (www.w3.org) has this definition:

> The *World Wide Web (Web)* is a network of information resources. The Web relies on three mechanisms to make these resources readily available to the widest possible audience:
> 1. A uniform naming scheme for locating resources on the Web (e.g., URLs).
> 2. Protocols, for access to named resources over the Web (e.g., HTTP).
> 3. Hypertext, for easy navigation among resources (e.g., HTML).

In other words, it means that you can find *things/stuff/information/data* (via URLs), retrieve it (via HTTP) and see it (via HTML). The explosive growth and extent of the Web has raised engineer's expectations of easy access to information. The Web is an excellent solution for engineers and technicians responsible for test, measurement, data acquisition or process control applications that generate information their colleagues need. Using the web, they can disseminate data they collected or present summary information from production results, process monitoring, etc. Also engineers can remotely check in on a test system, or remotely review measurement results from the latest experiments. This method of using the web to remotely monitor tests or data is called *"web monitoring"*.

Strictly speaking, the Web is a global, non-centralized network of computers operating with TCP/IP, HTTP, and HTML as a base set of computer protocols. However, removing the global aspect, Web technologies can also apply to local networks. Corporate intranets use the same Web technologies to build up a private network. Computer applications exchange information on corporate intranets with the same core set of protocols as those used over the World Wide Web.

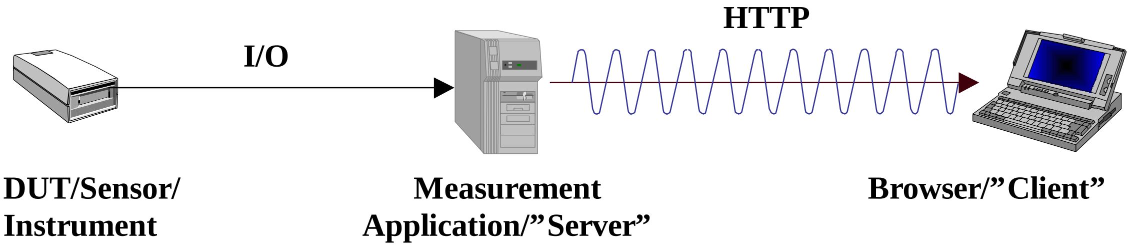
Measurement systems generate data that can be viewed with a web browser (such as Netscape or Internet Explorer). Programming languages such as Visual Basic and HP VEE can create test system programs as well as browser applications that enable web monitoring. Scripting languages such as VBScript allow the creation of more interactive HTML documents on the browser. Finally, engineers can easily create HTML documents for use on the Web with new enhancements to Microsoft Office97 products (Word, Excel & Access).

From one perspective, Web technologies simply offer another arena for development efforts. When a developer deploys ActiveX Components in conjunction with Web technologies, for example, the developer can go about it differently—incorporating HTML and Visual Basic Script code, providing security features, and so forth. But even with Web technology the developer still calls methods, sets properties, and handles events. In this way, all of the development team's knowledge about using Microsoft Visual Tools can be carried into the Web arena.

Overview of Web-Related Technologies

To simplify, let us assume you are creating a web site internal to your organization (also called an Intranet) and you want to provide data from your measurements to users, and those accessing the site already have graphical browsers.

A simplified model might look like this:



The item under consideration by the engineer acquiring data or testing products is the **Device Under Test, Sensor or Instrument**.

The **I/O Layer** includes the hardware interface such as GPIB, RS232, VXI, MXI, PC Plug-in (PCPI) cards as well as the I/O programming layer such as VISA, SICL, NI-488, DT-OpenLayers or others.

The **Measurement Application** is the software program (using languages such as Visual Basic, Visual C++, HP VEE or LabVIEW) that will generate the data and analysis of the phenomenon under consideration. It will be acting as the server of information in this example.

The HyperText Transfer Protocol (**HTTP**) is the computer communication protocol built on top of TCP/IP and allows communication between a server application (in this case our Measurement Application) and a client (in this case our Web Browser).

A user’s **Browser** (for instance Internet Explorer or Netscape Navigator), acting as the client, will be making a request of the information generated by the engineer’s measurement (server) application. The browser accesses an HTML document or file for information. If the document requires interaction, a scripting language or ActiveX document is required.

Web Communication Components

URL

Every resource available on the Web -- HTML document, image, video clip, program, etc. -- has an address that may be encoded by a *Universal Resource Locator*, or "URL".
A URL is usually entered in your browser to request information from a server. For instance to access HP VEE information on HP’s corporate web server you would type in a URL as follows:

http://www.hp.com/go/hpvee

This URL may be read as follows: There is a document available via the HTTP protocol residing on the machine www.hp.com, accessible via the path "/go/hpvee".

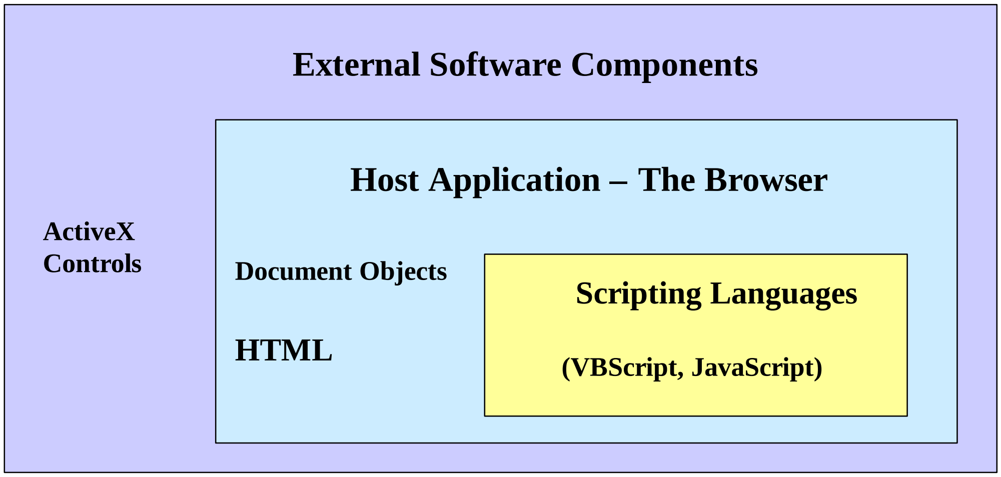
HTTP

Once a request has been made to the server it returns a stream of information in HyperText Markup Language (HTML) that your browser then interprets and displays to you. CAUTION: HTML is an *extremely* rapidly changing language and the World Wide Web Consortium has recently endorsed HTML version 4.0. This new version now includes more multimedia options, scripting languages, style sheets, better printing facilities, and documents that are more accessible to users with disabilities. There is a vast amount of information available on the latest standards for HTTP and HTML (as well as other ongoing standards work) at the World Wide Web Consortium’s web site: www.w3.org.

Scripts

A client-side *script* or *scripting language* is a “program” that may accompany an HTML document or be embedded directly in it. A scripting language is a browser-supported interpreted language that can extend the limitations of HTML to provide a more interactive Web page. Because they are interpreted, scripting languages must be embedded into the Web page and supported by the browser – they are not independent programs. The program executes on the client's machine when the document loads, or when some event occurs such as when a link is activated. HTML's support for scripts is independent of the scripting language. Scripts offer authors a means to extend HTML documents in highly active and interactive ways. Most scripting languages are limited for security reasons – that is they do not have the ability to use all systems resources. This is sometimes called *“sandboxing”* which means that the scripting language has a *"sandbox"* to play in and cannot go anywhere else.

Scripting Language Host Model



The scripting languages must reside inside a host application (i.e. the broswer). Within the browser, documents can be linked to and the overall look and feel of the web page is controlled by HTML. This is shown in the diagram below.

Some uses of scripts are:

- Scripts may be evaluated as a document loads to modify the contents of the document dynamically.
- Scripts may accompany a form to process input as it is entered. Designers may dynamically fill out parts of a form based on the values of other fields. They may also ensure that input data conforms to predetermined ranges of values, that fields are mutually consistent, etc.
- Scripts may be triggered by events that affect the document, such as loading, unloading, element focus, mouse movement, etc.
- Scripts may be linked to form controls (e.g., buttons) to produce graphical user interface elements.

Unfortunately there are various and rapidly changing script language "standards" available, generally promulgated by the creator of the browser. We have VBScript, Jscript by Microsoft, JavaScript support by Sun and NetScape and ECMAScript supported by the European Computer Manufacturers Association. We include here some information (supplied by the vendors) about what each of the scripting languages is. As you can see, they are all "fast", "portable", "lightweight", etc. And all different!

What is VBScript?

Visual Basic Scripting Edition (VBScript) is a subset of the Microsoft Visual Basic language. It is implemented as a fast, portable, lightweight interpreter for use in World Wide Web browsers and other applications that use ActiveX Controls, OLE Automation servers, and Java applets.

What is JScript?

Microsoft JScript is a powerful scripting language targetted specifically at the Internet. It is implemented as a fast, portable, lightweight interpreter for use in World Wide Web browsers and other applications that use ActiveX Controls, OLE Automation servers, and Java applets.

How does JScript compare to VBScript and Java?

When used in Internet Explorer, JScript is directly comparable to VBScript (not Java). Like VBScript, JScript is a pure interpreter that processes source code embedded directly in the HTML. JScript code, like VBScript code, does not produce stand-alone applets but is used to add intelligence and interactivity to HTML documents. For developers familiar with C and C++, JScript provides familiar syntax and language features.

What is JavaScript?

Netscape Navigator 2.0 and Netscape Navigator Gold 2.0 provide flexible, lightweight programmability via Netscape JavaScript, a programmable API that allows cross-platform scripting of events, objects, and actions. It lets page designers access events such as startups, exits, and user mouse clicks. Based on the Java language, JavaScript extends the programmatic capabilities of Netscape Navigator to a wide range of authors and is easy for anyone who can compose HTML. Use JavaScript to glue HTML, inline plug-ins, and Java applets to each other.

What is JavaBeans?

JavaBeans is a portable, platform-independent component model written in Java, developed in collaboration with industry leaders. It enables developers to write reusable components once and run them anywhere -- benefiting from the platform-independent power of Java. JavaBeans acts as a Bridge between proprietary component models and provides a seamless and powerful means for developers to build components that run in ActiveX container applications.

What is ECMAScript?

The ECMAScript standard describes a Web scripting language that can enrich and enliven Web pages in a Web browser. ECMAScript is the only standard scripting language on the Web; it is based on the ECMA-262 specification, which outlines an object-oriented programming language for performing computations and manipulating objects within a host environment, such as the browser. The complete ECMA-262 specification can be found at the ECMA homepage at http://www.ecma.ch

Currently, Internet Explorer will support both VBScript and or JScript. Version 4.0 of IE is claimed to understand ECMAScript. When the engineer creates the Web page front-end, instances of VBScript, JavaScript or both can be used for the Internet Explorer. For the Netscape Navigator, only instances of JavaScript can be used.

Automatic Browser Updating

In general browsers makes a request for the server information based on a person hitting the refresh button or some other kind of browser interaction. Browsers will not automatically update, unless the browser is designed to do so and the web page is designed to handle it. Also, there is not any interaction allowed by the browser unless it is designed into the browser page. The only way to do this is with a scripting language as mentioned above.

The Component Object Model and ActiveX Technologies

Component (or object) software applies the hardware reuse concept to the creation of new software. Until recently, approaches to software reuse have not been sufficient, either technically or in their widespread market acceptance. Current, traditional technologies present three obstacles to creating a component software market: 1) Distributing objects with their source code; 2) Reusing objects across different languages; and 3) Relinking or recompiling an entire application when one object changes.

To solve these problems and to move towards fulfilling the goal of true software component reuse, Microsoft architected the ***Component Object Model*** (COM). COM is a foundation for interaction among all kinds of software. COM defines a common way to access software services. Without COM there are many different mechanisms are used to access the services provided by libraries, local processes, the operating system, and remote processes.

What is ActiveX?

It is a marketing name for a set of technologies and services, all based on the Component Object Model (COM). It is also a set of *language-independent* interoperability technologies that enable software components that are written in different languages to work together in networked environments. Some of the benefits ActiveX components (from Microsoft) are 1) Versionable – there is a specification for determining which version of the object you are dealing with); 2) Programmable – via a set of interfaces you can "discover" via one standard way; 3) Lightweight and fast – although this is completely determined by the vendor writing the code; 4) An open standard (though controlled by Microsoft) which allows them to be written in any language and are 5) Networked via DCOM – DCOM is simply the networked version of COM.

Today, most COM-based technologies are assigned the label ActiveX. However, ActiveX is really a stripped down version of a COM object to make it more readily usable over the Internet. Currently COM is available for Windows95 & NT, Macintosh, and Solaris operating systems.

Microsoft’s Component Object Model (COM) “defines” compiled binary software components. **ActiveX Controls** based on COM can also be run in a browser. This topic is covered more fully in the white paper titled “*Understanding ActiveX…*”. Typically, an ActiveX Control is used to provide user interface functionality and is designed to run on the client computer. ActiveX components, unlike scripting languages, are able to use any system resources and as such are claimed to be “unsecured” by opposing viewpoints, i.e., vendors. They are often digitally *“signed”* by the company creating them to ensure that you are downloading the legitimate component. This does not make it any more secure on your system, it just means that you have received the correct component.

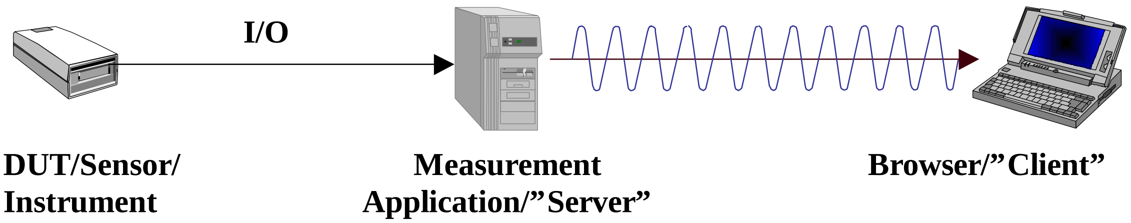
Extend Office 97 Applications

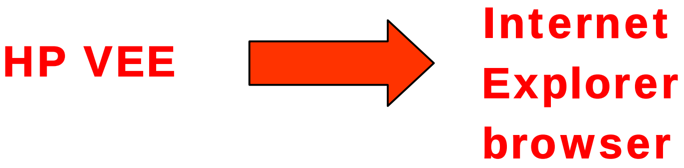
The ability to create customized business solutions with each of the Microsoft Office 97 applications is much easier through Office 97 as each of these applications have been made into very extensive ActiveX automation servers and their support Visual Basic for Applications (VBA). In addition, Microsoft is working with over 50 other companies on licensing Visual Basic for Applications. Visual Basic for Applications allows access to application functionality through ActiveX components. There is also a market for “VBA clone” development environments which are much less expensive to license.

Web Monitoring Using HP VEE 5.0

Server Measurement Application

HP VEE typically generates large quantities of data. Various individuals view the data or the results of the data. Test results, laboratory information, production efficiencies and quality reports are all examples of information that several people may want to view using the convenience of their web browser as shown below:

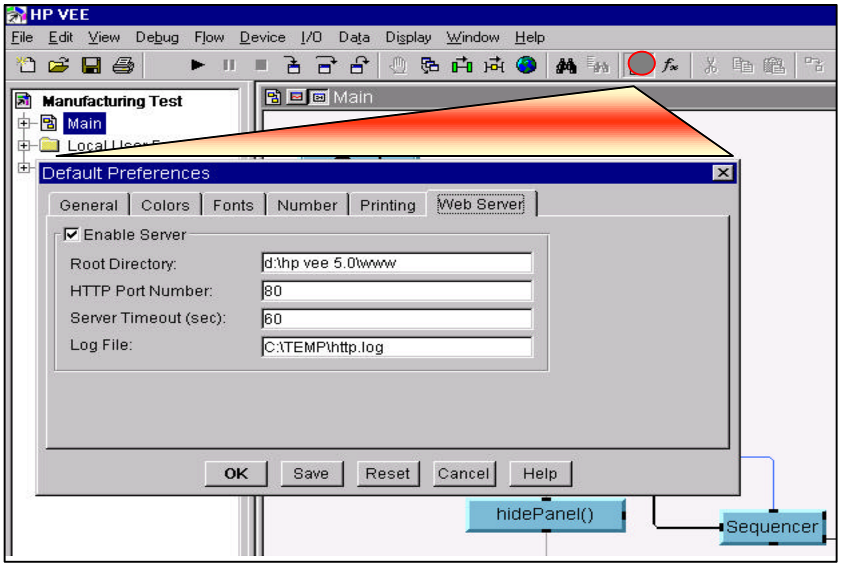




HP VEE 5.0 has the ability to allow a browser to see the view from parts of a VEE program, which means it has a built-in web server. When the designer wants to enable the server in HP VEE to allow data to be viewed, he will enable it via the new “**Web Server**” tab in Default Preferences dialog box. HP VEE must be running on a system that is connected to a network for the web feature to work.

Step One: Enabling HP VEE as a Web Server

The "Web Server" tab in the Default Preference dialog box looks like the following. (The Default Preference dialog box is popped up by the icon on the toolbar or from the *File→Default Preferences* menu pick.)


HP VEE
File Edit View Debug Flow Device I/O Data Display Window Help
Manufacturing Test
Main
Local User F
Main
Default Preferences
General Colors Fonts Number Printing Web Server
OK Save Reset Cancel Help
hidePanel()
Sequencer

☑ Enable Server

Root Directory: d:\hp vee 5.0\www

HTTP Port Number: 80

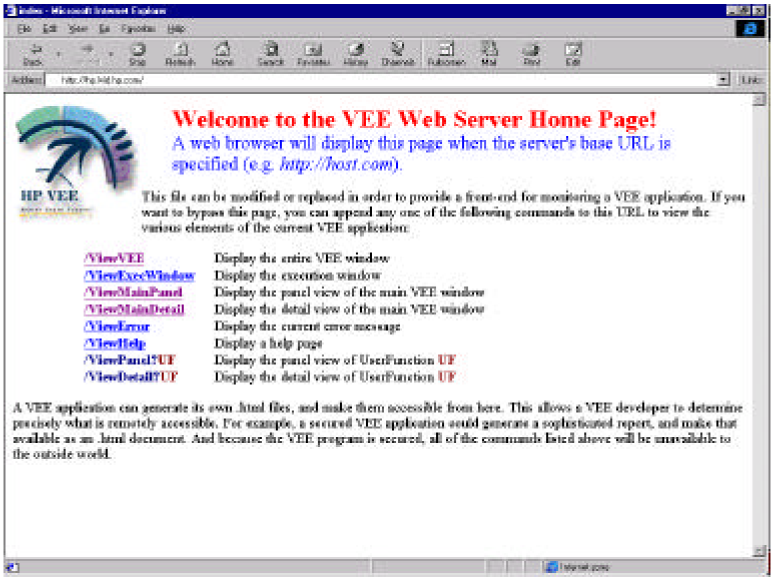
Server Timeout (sec): 60

Log File: C:\TEMP\http.log

If HP VEE 5.0 is installed on a Win95/NT operating system with the default suggestions, this field will read "C:\Program Files\Hewlett-Packard\VEE 5.0\www." By changing the HTTP port the server "listens on" from the default of 80, you can limit who can see the data as only they will know the correct port on which to send the data from their browswers. For instance if port 91 were chosen the user would use the following URL: http://hostname:91.

Step Two: Viewing the VEE Program in the Client Browser

When a "default" request to the VEE web server is received VEE will send the default web page back to the browser called *index.html* (found in the "www" directory in the VEE installation directory hierarchy) and it looks something like the following:


index - Microsoft Internet Explorer
File Edit View Go Favorites Help
Back Forward Stop Refresh Home Search Favorites History Channels Fullscreen Mail Print Edit
Address http://hp.hld.hp.com/
Links
HP VEE
Internet zone

Welcome to the VEE Web Server Home Page!

A web browser will display this page when the server's base URL is specified (e.g. *http://host.com*).

This file can be modified or replaced in order to provide a front-end for monitoring a VEE application. If you want to bypass this page, you can append any one of the following commands to this URL to view the various elements of the current VEE application:

/ViewVEE	Display the entire VEE window
/ViewExecWindow	Display the execution window
/ViewMainPanel	Display the panel view of the main VEE window
/ViewMainDetail	Display the detail view of the main VEE window
/ViewError	Display the current error message
/ViewHelp	Display a help page
/ViewPanel?UF	Display the panel view of UserFunction UF
/ViewDetail?UF	Display the detail view of UserFunction UF

A VEE application can generate its own .html files, and make them accessible from here. This allows a VEE developer to determine precisely what is remotely accessible. For example, a secured VEE application could generate a sophisticated report, and make that available as an .html document. And because the VEE program is secured, all of the commands listed above will be unavailable to the outside world.

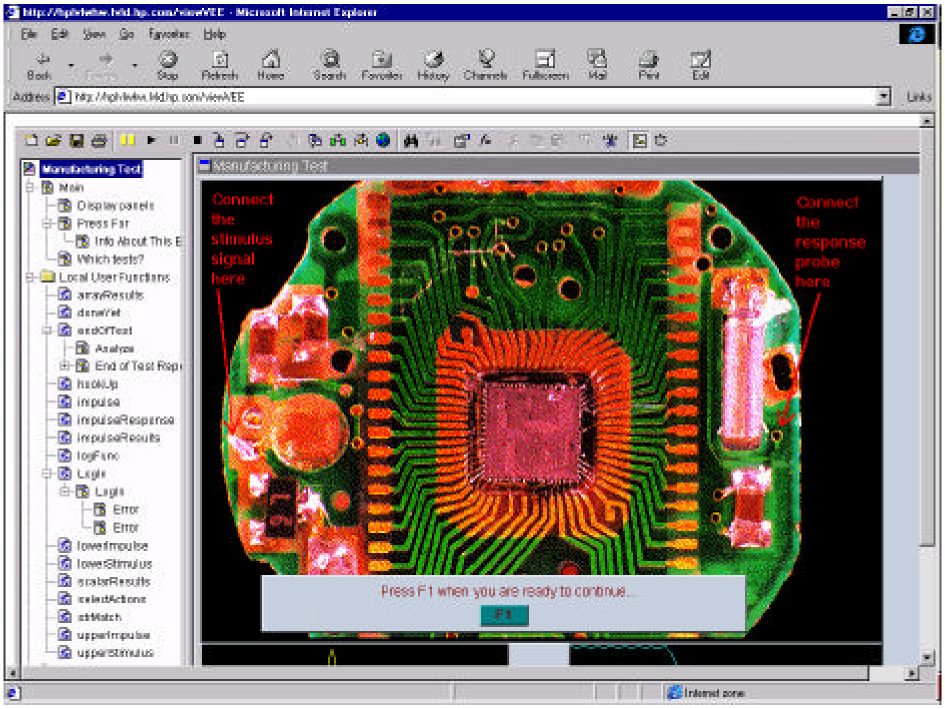
This web page defines the commands that the VEE web server will accept. So you can navigate the VEE program by clicking on the appropriate command on the web page shown above, or directly type in the URLs as:

http://hostname{:port}{/file}
or
http://hostname{:port}{/command}{?parameter}

Where:
hostname (required) identifies the host system on which HP VEE is running. An example might be: "hplvlhob.lvld.hp.com."
port (optional) identifies the Web server port number if not using the default value 80. This value must match the port number specified on the Web Server tab in Default Preferences.
file (optional) identifies a directory and/or file relative to the root directory for the browser to open. You can include a file name extension, such as .jpeg or .html). You must specify the root directory on the Web Server tab in Default Preferences.
command (optional) specifies a command supported by the Web server letting you monitor, troubleshoot, and control an HP VEE program from the browser. Commands are described below.
parameter (optional) specifies a parameter required by a command. Parameters are described with the Commands below.
After entering http://hostname/, the user can do any of the following:

Action	**URL to use**
Change the default port number	http://hostname:portNumber
Accessing the non-default file "page.html"	http://hostname/page.html
To View	**Add after http://hostname/**
Entire VEE window (titlebar, menu, etc)	ViewVEE
VEE Main Panel View (user interface of Main)	ViewMainPanel
VEE Main Detail View (program code in Main)	ViewMainDetail
VEE execution window during runtime	ViewExecWindow
Panel view of a UserFunction (user interface of UF)	ViewPanel?UserFunctionName
Detail view of UserFunction (program code in UF)	ViewDetail?UserFunctionName
Error window of current program	ViewError

An example from an Internet Explorer shows how these commands will display the program shown earlier. For instance, by using the command: http://hp.lvld.hp.com/ViewVEE with the "mfgtest.vee" program from the …\examples\apps\ directory, the user might see results in their browser like the following:



http://hplvlwhw.lvld.hp.com/viewVEE - Microsoft Internet Explorer

Address http://hplvlwhw.lvld.hp.com/viewVEE

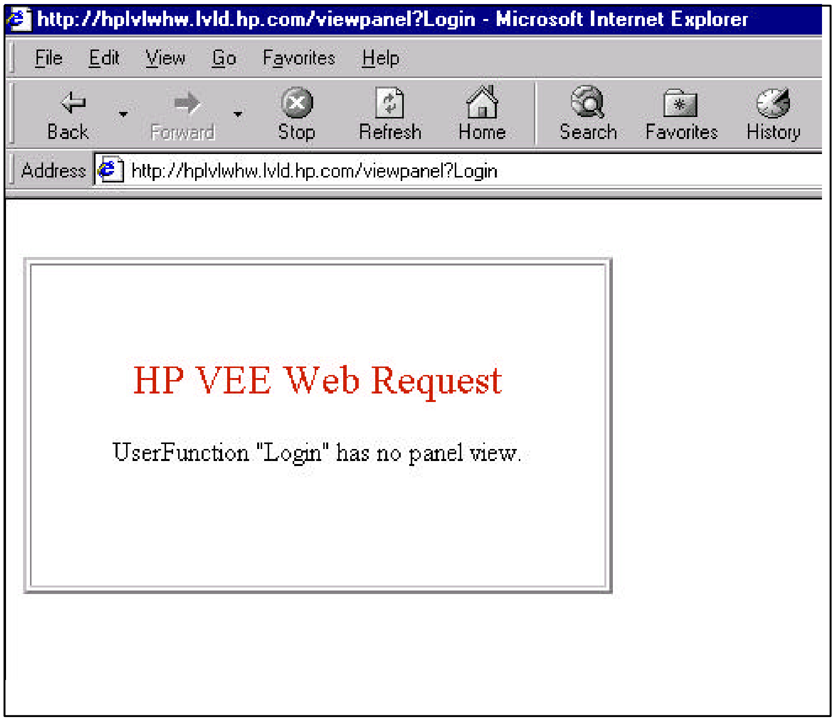
Manufacturing Test

Connect the stimulus signal here

Connect the response probe here

Press F1 when you are ready to continue.

If the user wants others to see only the user interface, then the command http://hp.lvld.hp.com/ViewMainPanel is used. But let us assume the program did not have any user interfaces put on the panel view the user sees. This results in the following display:



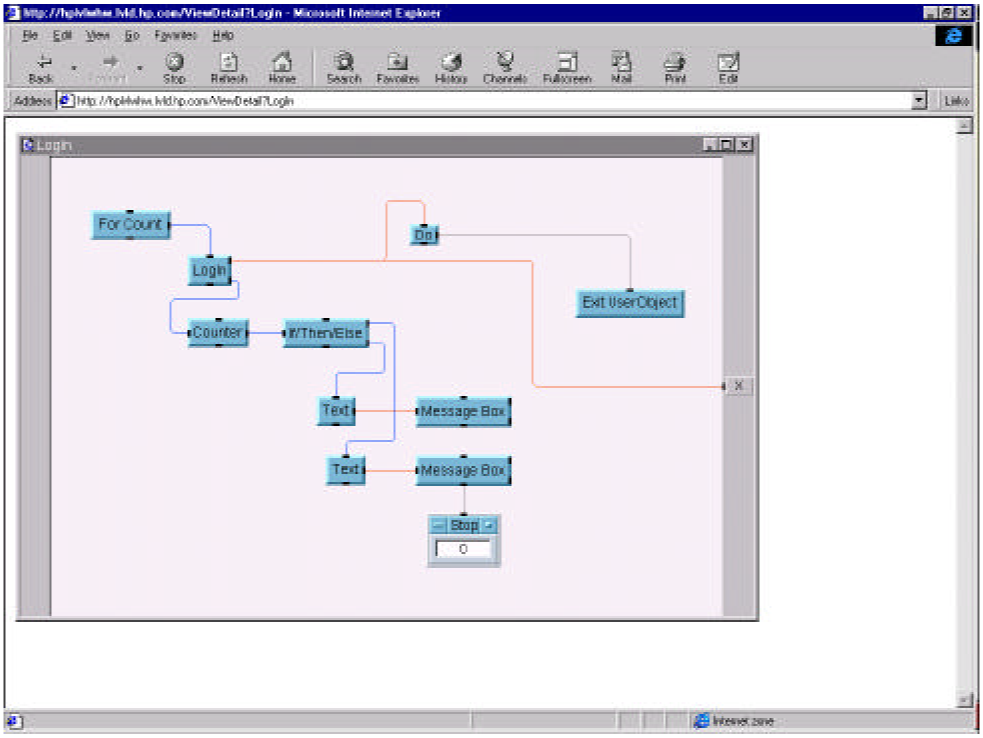
http://hplvlwhw.lvld.hp.com/viewpanel?Login - Microsoft Internet Explorer

Address http://hplvlwhw.lvld.hp.com/viewpanel?Login

HP VEE Web Request

UserFunction "Login" has no panel view.

If an engineer wants another to see his programming code, those viewing the HP VEE program can access the code of the Main program or any User Functions by using the ViewMainDetail command or the ViewDetail?UserFunction command. Below is an example of User Function called "Login" from the manufacturing test example from above. The user types in: http://hp.lvld.hp.com/ViewDetail?Login:



As you can see, this allows several engineers to troubleshoot code remotely by looking at the details of the HP VEE program code loaded on another machine.

Security

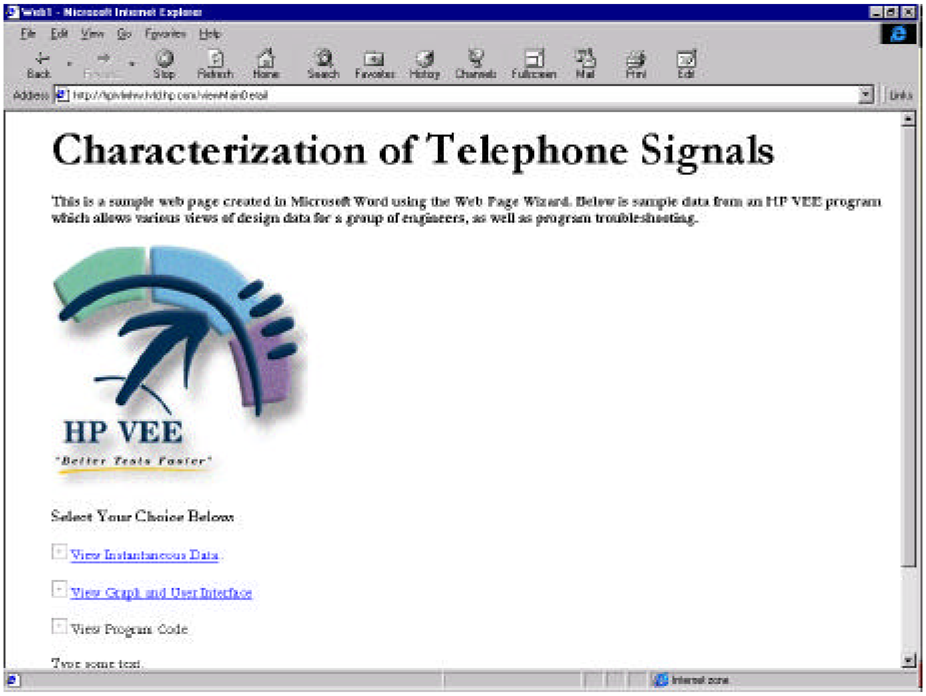
However, security might be a concern for the engineer. There are numerous technologies that address an internet solution which is beyond the scope of this paper. This paper discusses an intranet application (i.e. within one company or organization). There are a couple of ways to make it difficult to view information from a VEE Web Server in an intranet application:

1) Change the port number in the *Default Preferences* → *Web Server* Tab so only those who know the port number may view the program.

2) Create an HTML file with the exact name of the commands that are to be disabled and put the page in the HP VEE “www” directory. HTML files can be created easily in Word97, and several templates and wizards exist to help speed up the process. For more polished and complicated HTML files, use programs like Microsoft's FrontPage.

If the file exists, the VEE web server will send that file back to the requester instead of displaying the view of the requested function. In this way, the creator of the HP VEE program can intercept those requests and display a HTML page with the appropriate warning or comments.

For example, an HTML file named "ViewDetail" was placed in the "www" directory of the HP VEE file structure. So when the URL http://hp.lvld.hp.com/ViewDetail?Login is initiated, the following results:


Web1 - Microsoft Internet Explorer
File Edit View Go Favorites Help
Back
Forward
Stop
Refresh
Home
Search
Favorites
History
Channels
Fullscreen
Mail
Print
Edit
Address http://hpvlview.lvld.hp.com/viewMainDetail
Links
Internet zone

Characterization of Telephone Signals

This is a sample web page created in Microsoft Word using the Web Page Wizard. Below is sample data from an HP VEE program which allows various views of design data for a group of engineers, as well as program troubleshooting.

Select Your Choice Below:

View Instantaneous Data

View Graph and User Interface

View Program Code

Type some text.

3) A secured runtime version of the HP VEE program can be created. This will insure that, at least none of the program code will be viewed.

Completing the Distributed Measurement Application

If the data is only meant to be monitored, i.e., the user viewing the data on the browser is not meant to input information or make choices that impact what is viewed, then the examples shown above are complete. However, if the application requires interaction over the web, additional programming is required.

The options available for building interactive web pages include VBScript (with higher level development environments such as Microsoft's FrontPage), Visual Basic programs, and ActiveX documents.

Scripting languages provide a way to create interaction with a Web page. Engineers can calculate results, provide feedback, initiate data storage. In addition, scripting languages can use intrinsic HTML forma controls and ActiveX controls to provide a Web page with "look and feel." Also available from the scripting language via ActiveX automation is the ability to tie other applications into the Web page.

There are drawbacks to scripting languages. For instance, VBScript does not have Visual Basic's *Timer* function. It only uses the VARIANT data types. VBScript also has less control flow/conditional capabilities – the engineer can not construct an *If…Then…Else, DoEvents, For Each… Next, GoSub… Return, GoTo; On Error GoTo; or On… GoSub, On…GoTo.*

For continuous monitoring of data, continuous "snapshots" of the HP VEE program are required. The simple case demonstrated in this paper requires the user to manually hit "refresh" in their browser when new data is requested. If automatic updating is desired it is possible to us a JavaScript or possibly VBScript program to continually refresh the browser screen, probably in the order of 5-10 times per second maximum. This is beyond the scope of this white paper.

Summary

With HP VEE 5.0's web server capabilities, you can easily make your measurement application, and the resulting information, available for other interested individuals. Test results, live data, experiments in progress, production yields, or quality data may be quickly viewed through a common user interface -- the browser. HP VEE's web server capabilities make it easy to remotely monitor any part of a HP VEE program. Program code may be examined remotely for joint discussed on program development. There are many software tools available to enhance the browser interface to allow interaction with the measurement data presented. These capabilites combined with HP VEE's ability to be an ActiveX Container allow the engineer the ability to easily customize their solutions in a standard environment.