

## The MMI: Web-Enabled Remote Access and Monitoring of Equipment

### Introduction

The World Wide Web provides an ideal GUI (Graphical User Interface) for machine and sensor MMI (Man-machine interfaces). Because of its standardized and portable nature, various components of the Web allow for direct access to information from a wide variety of computing platforms, from desktop PCs to cell phones. Web Page designers can embed programs and algorithms into the pages themselves, and servers can communicate directly with embedded applications. This combination enables complex data-driven pages to present essential information without information overload. Achieving all these goals without custom application development makes the Web an ideal platform.

### Weaving the Web

There are several components of a web-based application: The Web server provides the data requested by the client; HTTP (Hypertext Transport Protocol) is the protocol used to transport requests and data between the server and the client; HTML (Hypertext Markup Language) is the language and syntax for the content of web pages; The Browser is the client component that runs on a users' computer.

We often picture a Web Server in a roomful of computer and networking equipment, manned by a fleet of I.S. managers. While that is true of general purpose Web Servers, there is a trend toward embedding application-specific Web Servers into networking devices. Most networking devices (routers, bridges, access points), have web-based interfaces for configuration. An embedded Web Server allows any standard browser to gain access to the controls of the device.

HTTP is utilized by the Web Server when requesting data. This standardization enables applications developers to design servers and clients that communicate interchangeably. HTTP defines HOW data is exchanged between Server and Browser.

HTML is an ASCII-based language that defines the content of the data exchanged between Web Server and Browser (data tags, templates and style guides). This standard also defines graphics and third-party applications such as audio and video.



Embed DPAC's Airborne™ 802.11b Wireless LAN Node Module into OEM applications in as little as 6 weeks.

- Highly integrated 802.11b wireless module with radio, base-band & application processor
- Built-in web server enables drop-in LAN and Internet connectivity
- Quick time-to-market & reduced development costs
- Reduces need for RF and communications expertise
- Integrated RTOS and TCP/IP Stack
- Configurable serial, digital & analog I/O ports

For more information, visit us at [http://www.dpactech.com/wireless\\_dsp.asp](http://www.dpactech.com/wireless_dsp.asp)

The chief benefit of using a browser-based interface is its portability. There are a number of browsers available for just about any computing platform. This applies to desktop and notebook computers as well as handheld computers, PDAs, and cell phones. This allows access to the data on a Web Server from literally anywhere in the world, at any time, thanks to the standardized components. This also means that there are no application-specific programs to install to gain access to device-specific data

## Turning Pages

There are typically three types of Web Pages: Static pages represent unchanging information; Dynamically built pages are used to present data that change frequently; and Hybrid pages that contain both static and dynamic data. Proper selection of page types can minimize the Web Site programming effort.

Static pages are best used to present content that doesn't change at all, such as basic equipment information. Such content includes installation, operation or maintenance information. Consider a device whose diagnostic manual is in static pages, local to the device, where a technician can use a handheld computer to walk through diagnosis and repair of the equipment. Static pages are built using a simple HTML editor.

Dynamic content is best presented on a "fill in the blanks" type of page. Examples include tables of labels and data, where the labels are statically defined, but the content is retrieved from the device. A Computer Graphics Interface (CGI) is a device-independent language typically used to retrieve the parameters from the device and present them to the Web Server to integrate into the page.

Sometimes dynamic content isn't enough, and more flexibility is required in the page. Dynamic pages allow an interaction between the browser and the device that can create new pages on the fly, without changing the HTML description of the page. These pages use scripting languages like JavaScript, or interpreted languages like Java to allow the browser to "POST" specific requests to the Web Server for additional information. The Web Server provides the JavaScript or Java just like any other page. The advantage of having the browser execute the script to create the page content is that it allows very simple Web Servers and devices to create very complex pages without rewriting internal executable code. This allows interaction, conditional display of the page content, content-defined lists and even complex graphical representation

## Publish or Perish

Programs and procedures can be embedded into Web Pages using (typically) Java or JavaScript. Java and JavaScript are fairly unrelated, other than both being derived from the C language. The advantage of both Java and JavaScript is that they execute on the host computer in the browser environment. CGI applications provide dynamic mechanisms that execute on the server. These mechanisms allow the server to create dynamic pages and interactions without providing code to the client.

The code for JavaScript is embedded in web pages, so the developer can see the actual JavaScript source code. The browser itself interprets JavaScript and the effects are created on the browser screen. The primary advantage of JavaScript is that the code is maintained in ASCII and can be completely embedded within an HTML page. Like any other HTML page, it can also be a file externally referenced by an HTML page.

Java is more of a compiled language. The Java source code is compiled into an intermediate code that is interpreted by a plug-in program in the browser. The plug-in is a separate, stand-alone operating environment running on the computer. The primary advantage is that the source code is not available to just anyone that wants to look at it, providing a security mechanism between the Java applet and host computer, so applets can run securely on insecure platforms (such as cell phones).

The CGI is a defined interface between the server and other applications running on the server. It allows a number of differing programs to run or be invoked by a Web Page, and the results provided to the client. There is no language dependence, so a Web Page can invoke a CGI call to a compiled C/C++/C# program, or interpreted Perl, PHP, Python, Java, or any of a number of compiled or portable languages. Since CGI applications run on the server, their specifics are kept hidden from the browser.

For embedded systems, JavaScript and Java are the more prevalent ways to create dynamic Web content. Because they fit neatly within the file-based constructs of HTTP, and do not require custom code, they can easily be customized and loaded on the Web Server without re-compiling the embedded application. This allows OEMs and end-users to customize the MMI without regard to the specifics of the platform the server runs on.

## Conclusion

Because of the capabilities of browsers to provide not only text, but graphics as well, complete MMI interfaces can be designed for embedded sensors. Since the MMI goes with the sensor, rather than the workstation, these interfaces allow technicians to directly view the settings and status of equipment from a wide variety of platforms. The flexibility of the Web architecture allows complex customization of pages without re-writing device firmware. All of these benefits lead to a powerful MMI architecture ideal for factory and device status and configuration monitoring.