Press release
Real Time Logic

Real Time Logic Announces High-Performance Embedded Web Server For the QNX(r) Neutrino(r) real-time operating system

RealTime Logic's Barracuda web server simplifies remote real-time monitoring and control of distributed embedded systems

April 3, 2006 - Real Time Logic, the world leader in device management software for telecommunications, medical and automotive applications, today announced the availability of the industry's most powerful embedded web server for the QNX real-time operating system. The web server makes it easy to remotely monitor, control, and configure embedded applications running on the QNX real-time operating system. Barracuda is particularly effective for complex distributed systems, enabling browser clients to monitor and control multiple devices at the same time with full authentication, authorization, and encryption facilities.

"Barracuda is an industrial strength embedded web server, optimized for sophisticated distributed processing applications," said Shannon Dougall, director of product marketing at QNX. "Barracuda extends the distributed processing capabilities of the QNX real-time operating system, making it easy to monitor and control multiple QNX nodes in real time from the same remote browser."

Barracuda utilizes the standard HTTP protocol to facilitate secure transfers of user data, firmware, and application control data between clients and servers. Unlike conventional HTTP-based clients, Barracuda provides an EventHandler that enables HTTP- and Java-based clients to monitor and control multiple event-driven embedded applications residing on multiple devices in real time. The full-duplex, asynchronous EventHandler protocol stack, which sits on top of HTTP, gives EventHandler clients the same benefits as regular HTTP clients, including the ability to bypass proxies and firewalls.

The EventHandler uses a compact (32 kbytes), real-time, Secure Socket Layer stack (SharkSSL), which supports hardware crypto encoding and decoding for common hardware crypto engines, to establish secure client-server communications.

Barracuda's EventHandler features install-on-demand technology that enhances device efficiency and reduces memory requirements by minimizing the amount of application code that must be stored in local device memory. With Barracuda, users can store their device application code on a remote server. When the user activates an URL to an EventHandler-based application, Barracuda installs the DHTML browser application on demand. Barracuda also supports Sun's Web Start Technology, which enables remotely hosted Java applications to be installed on demand, in the same way that Java applets are downloaded on demand to PCs.

Barracuda's CSP language (similar to Active Server Pages) makes it easy to generate traditional server web-applications. With CSP, developers can embed C or C++ programs directly into an HTML template page created by an HTML designer. The rich C and C++ API, modeled after enterprise server APIs, provides methods that make it easy to design applications based on the common Model-View-Controller paradigm. The object oriented virtual file system makes it possible to delegate user requests, such as HTTP GET and HTTP POST, to any resource or resource collection.

Barracuda comes with a number of ready to use resource collections, including the ability for clients to read and write files to any directory in an embedded file system, and read files directly from within ZIP files. Resources can be dynamically installed and removed during runtime. Thus, a ZIP file can be uploaded using HTTP(S) and dynamically installed on the virtual file system.

Barracuda provides a number of advanced plug-ins, including web services, such as SOAP, and a web interface to any SNMP MIB. Barracuda's host tools include a graphical MIB editor and compiler, Interface Definition Language (IDL) compiler for the EventHandler, and a compiler/linker for the CSP scripting language. Barracuda also provides a host simulation environment that enables designers to develop, run and test their code on any Unix or Windows machine without requiring target hardware.