

Easily Integrating World with WebSphere

By Jeff Geiger

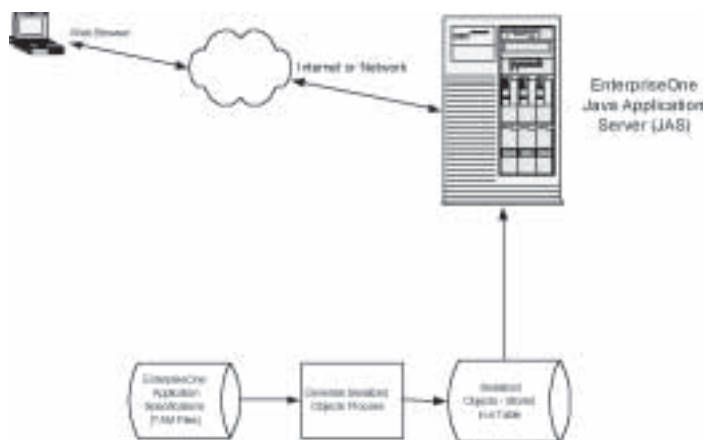
With an incredible investment in World software and other legacy systems, companies are forced to look at other means for getting their applications to the web besides upgrading to or replacing the existing system with a web-enabled ERP system such as EnterpriseOne. Collaboration is the name of the game when it comes to linking suppliers, customers, employees and the existing system into one seamless, integrated unit.

Often times, the same companies that have significant time and effort invested in the legacy systems in terms of hardware, software, data and know-how, also have significant sweat invested in custom systems and custom code that gives them competitive advantage. Even though there is customization, they still have the need to turn on a dime as their customers require them to and their competitors force them to. In this article, we will discuss re-use of your legacy system assets for a fictitious collaborative customer-facing system.

First of all, let me clarify that I am not going to get into the technical details of installing, configuring, programming and implementing an IBM WebSphere solution. I am merely going to talk conceptually about what components need to be in place and what can be done. Finally, we will discuss, candidly, how you can justify the efforts of such a project.

Overview of WebSphere

WebSphere is a general term to describe a broad range of e-business enabling products and services. It is at the heart of deploying JD Edwards EnterpriseOne to the web. J.D. Edwards spent a considerable amount of time and money developing the process for generating JAVA enabled components and the tools to publish applications to the web without requiring the overhead of another development environment. It looks something like this:



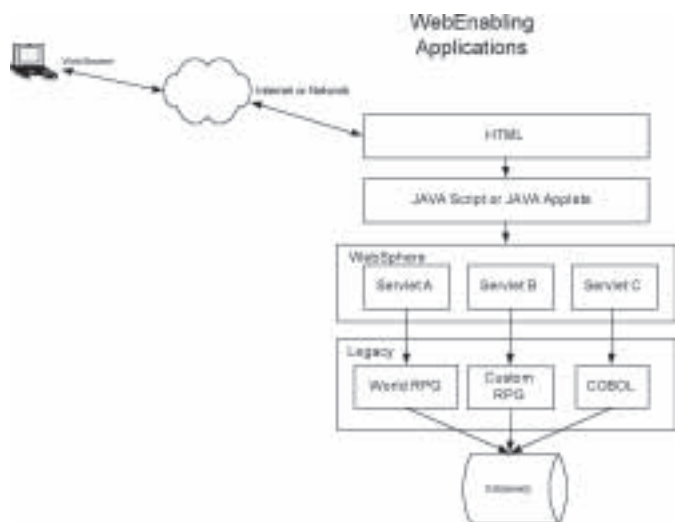
The Java Application Server (JAS) is where a combination of tools and technology is installed including EnterpriseOne JAS Software, WebSphere Application Server and IBM HTTP Server. Conceptually, it works like this: A web page (HTML and JAVA applets) is displayed to the user based on the serialized object.

Some processing happens on the web page when the user performs an action, the result then gets sent back to the JAS server where the requirement is interpreted and various processes occur including data input/output and business functions being called. The business functions run on the Enterprise Server. That's the same place the code would run if you were running in a Terminal Server environment.

Web Enabling

Enough "techy" information. To accomplish this same task in a non-EnterpriseOne environment, you must duplicate the overall scheme of the EnterpriseOne JAS server. That means that you need HTTP, JAVA and an interface to the legacy system (WebSphere).

Without rewriting your legacy investments, you can interface one or more applications to the web, even combining multiple processes from separate systems into one seamless process. That new process looks something like this:



How it Works

IBM has given us many tools to integrate applications and expose them to the web. These include WebSphere, Host Access Translation Services (HATS), Host Access Translation Services Limited Edition (HATS LE) and IBM WebFacing Tool. Each has its own benefits and costs. For the purposes of this discussion we are only going to discuss the single option of using WebSphere.

As the diagram shows, you can interface one or more applications to the web on one or more web pages. That means that with the IBM interface tools you can port your existing applications to the web even if they are on completely separate systems using different tools. This is very exciting because you can satisfy your needs for getting to the web with critical business processes without re-implementation.

Justifying the Expenditure

By implementing a WebSphere server, creating an HTML and JAVA front end with WebSphere Servlets to call the legacy business logic, you can move your applications to the web in weeks as opposed to months (or years) of upgrades required to get to a web-enabled version of the application you are using.

Part of the savings is the fact that you do not need to rethink the business logic, but merely to “call” it or “expose” it to the web with WebSphere. You can also keep the internal systems working as-is, but put a new face on them for anything external. You can keep the stoic applications inside, but have a cutting-edge look on the outside. This is your way to have your cake and to eat it too.

Conclusion

The concept of exposing your business logic to the web is not a new one, but it has taken companies a while to grasp just what you can accomplish and how quickly you can accomplish it by keeping your legacy code and avoiding an implementation or re-implementation. Choosing the right equipment, software, implementation partner and approach will mean success on your project.

Before deciding on tools and system architecture, work through the list of pros and cons and evaluate the return on investment with your IBM Business Partner and their IBM Representative. You’ll find that some tools interface quickly and easily, but have limitations or licensing issues while others may have higher up front costs, but are configurable and broad licensing.

Jeff Geiger is the Director of Technology for Oracular Minnesota, LLC, and can be reached at jgeiger@oracular.com.