



Patricia Seybold Group

Trusted Advisors to Customer-Centric Executives

IBM WebSphere Portlet Factory 5.11.3

Portlet Development through Component
Assembly and Mass Customization

By Mitchell I. Kramer

Sr. VP and Sr. Consultant, Patricia Seybold Group

IBM Corporation is authorized to distribute this report internally within IBM only.

Unauthorized redistribution of this report is a violation of copyright law. Direct link: <http://dx.doi.org/10.1571/pr10-13-05cc>

210 Commercial Street, Boston, MA 02109 • Phone 617.742.5200 • Fax 617.742.1028 • www.psgroup.com



IBM WebSphere Portlet Factory 5.11.3

Portlet Development through Component Assembly and Mass Customization

By Mitchell I. Kramer, Sr. VP and Sr. Consultant, Patricia Seybold Group

March 7, 2006

NETTING IT OUT

Customer portals make it easier for your customers to do business with you. They're portals that deliver your online customer experience. They support all of your customers' Customer Scenarios[®], providing access to all the resources that your customers need in one convenient, custom-tailored environment.

One of the most important yet most complex tasks in building customer portals is developing the portlets that implement and deliver applications services to your customers. Portlet development has required the same skills, techniques, and tools needed for all Web application development, and for integrating Web applications with back-office and external applications and data.

WebSphere Portlet Factory is a portlet development toolset that can simplify and speed portlet development. It lets your developers use component assembly and mass customization techniques to build portlets. This product was originally developed by Bowstreet, Inc., and introduced in 2002 as Bowstreet Portlet Factory. Bowstreet was acquired by IBM on December 20, 2005 and the product has been re-branded as WebSphere Portlet Factory in March 2006. To date, approximately 200 customers have installed the product. WebSphere Portlet Factory Version 5.11.3, introduced in December 2005, is its current release.

We recommend that you add WebSphere Portlet Factory to the toolkit of your portal platform. It's a viable product from an established growing company. In this report, we give you the justification for that recommendation.

WHAT ARE CUSTOMER PORTALS?

Customer portals are portals specially designed to make it easy for your customers to do business with you on the Web. They deliver all the portal services of conventional, general-purpose portals¹ but are specialized for your customer constituency, as follows:

- The audience for customer portals is a specific group of customers (e.g., small business customers or individuals in a specific client account).
- Customer portals are an integral component of the cross-channel, cross-lifecycle customer experience that you deliver. They support the Web channel, and they support activities in every phase of the customer lifecycle.
- The activities supported by customer portals are your customers' key Customer Scenarios[®], the sequences of activities that customers perform to achieve their objectives. We say that customer portals *contain* your customers' key Customer Scenarios[®].
- The key activities that customers perform through customer portals are accessing their own account information, their order information,

¹ General-purpose portals: 1) aggregate and deliver information from a variety of internal and external resources, 2) wrap up and present application functionality from existing internal IT systems and from external applications, 3) enable single sign-on authentication for the applications and resources that end users access via the portal, and 4) offer customization capabilities to portal owners and basic personalization capabilities to end users.

their incident/case information, and their contracts and service entitlement information.

- Role-based portal access (via single sign-on) determines the particular Customer Scenarios[®] that are featured on the portal.

While making it easier for your customers to do business with you, customer portals also have significant advantages for you. They can help you save money by enabling customers to serve themselves and to up-sell and cross-sell themselves. They can help increase customer satisfaction, loyalty, and profitability by providing everything the customer needs in one convenient, custom-tailored environment. And customer portals help you gain visibility into customers' account information, their needs, and their preferences.

Portal Technology Platforms Make It Faster and Easier to Design and Deploy Customer Portals...

You can build your customer portals from Web infrastructure, search, and content management technologies, but portal technology platforms make it easier to design, develop, deploy, and maintain customer portals. These platforms package facilities and tools that give you a big boost in building portals. Today's portal platform technologies now make it easier than ever before to design, develop, deploy, and manage hundreds or even thousands of role-based, customizable customer portals. Packaged in portal technology platform products are key components of customer portals. And you may not need skilled developers to build your portals. Many of these facilities have been designed for business users. (Really—it's not just marketing hype!)

...But Portlet Development Slows Deployment

One of the most important yet complex tasks in building customer portals is developing the portlets that implement and deliver application services to your customers. Portlets may access your existing application portfolio for these services, for example, letting a customer renew a maintenance agreement. They may provide data access, for example, letting a

customer see his order history. Or, they may provide portal-specific services, for example, creating a collaborative workspace in which customers can co-design the custom products that they want to buy.

So building portlets takes a combination of development, to create a UI and process customer requests and responses, and integration, to access applications and/or data. Most times, your staff is going to be doing that work.

Yes, it's true that portal platforms package many pre-built portlets, and additional pre-built portlets are available from the portal platform supplier's partners. But, like most application software, you'll need to customize these pre-built portlets to fit your application services requirements. Customization means programming (and it often takes longer to figure out the portlet's code and modify it than it does to start building the portlet from scratch).

Portal platform suppliers know that you'll have to do some programming to deploy your portals. Their platforms also package development tools for building and customizing portlets. Most frequently, these tools are extensions or add-ins to Java IDEs (independent development environments) that integrate the platform's portlet interfaces and structures. They let your developers build portlets with the same tools that your developers use for building other applications.

But, you typically need skilled Java developers to build portlets. Portal platforms are missing tools that let lesser technically-skilled folks, like business analysts, build portlets. Portal platforms are also missing mechanisms that improve the productivity of the portlet development process. So, portlet development is almost always both the critical path element and the highest cost element in the projects that implement your customer portals.

WebSphere Portlet Factory Simplifies and Speeds Portlet Development

WebSphere Portlet Factory is a portlet development toolset that simplifies and speeds portlet development. It gives developers of all skill levels mechanisms, built on patented technology, to build more portlets more quickly, to refine and improve portlet functionality iteratively and continuously, and to customize portlet functionality to the requirements of users and customers in different roles.

With WebSphere Portlet Factory, developers use component assembly and mass customization techniques to develop portlets.

This product was originally developed by Bowstreet, Inc., a software supplier founded in 1998 in Tewksbury, Massachusetts. It was introduced in 2002 as Bowstreet Portlet Factory. Bowstreet was acquired by IBM on December 20, 2005 and the product has been re-branded as WebSphere Portlet Factory in March 2006.

WebSphere Portlet Factory has an installed base of approximately 200 customers, more than half of which are Global 2000 companies. Most of WebSphere Portlet Factory's implementations have been for both internal and external portals, on the IBM WebSphere Portal platform. Version 5.11.3, introduced in December 2005, is the current release.

A FRAMEWORK FOR EVALUATING PORTAL TECHNOLOGY PLATFORMS

Evaluation Framework for Customer Portals

We've built a framework that can help you select the portal technology platform that best addresses your customers' requirements—the platform that best supports the content, data, and applications that your customers want to access to perform the activities of their Customer Scenarios[®]. The framework enables you to make apples-to-apples comparisons of the most important product evaluation factors. Then our product evaluations against the framework speed and simplify your work even more.

The framework for customer portal platforms has five top-level evaluation criteria, with sets of subcriteria for each top-level criterion. The top-level criteria are:

- Technologies for Supporting Customer Scenarios[®]
- Analytic Functionality
- Architecture
- Product Viability
- Company Viability

The subcriteria for the technologies for supporting Customer Scenarios[®] criteria are portlets, customer profile management, process management, content management, search, personalization, and collaboration. Portlets are the key subcriterion. They let you support Customer Scenarios[®] like:

- I want to go to one place to do all the tasks that I need to perform.
- I want to access, view, and change the customer information that you keep about me (and my company).
- I want to view the status of my current account profile (orders, transactions, portfolio, itinerary), the details of my orders, and my account history.
- I want to view the status of my open incidents and/or your work in progress on my behalf.
- I want to view my contracts, service entitlements, coverage, and/or benefits, and I want to see recommendations that are appropriate to me. (And if my company has identified “approved” or “standard” items to buy from you, or special pricing, I want that to be clear when I'm looking in your catalog.)

In our platform evaluations, portlets are the first subcriterion that we evaluate and the subcriterion where we've been finding the most differentiation.

Evaluating WebSphere Portlet Factory

Because WebSphere Portlet Factory is not a portal technology platform, we can't evaluate it against all of the criteria in our framework. However, it is a product that we think you should consider in addition to a portal technology platform for creating your customer portals. Its tools and facilities will help you develop and deploy portlets that deliver a broad range of services critical to your customer experience. So, in this report we'll evaluate it against the appropriate criteria and subcriteria of our framework—portlets. And, to help in your decision to acquire WebSphere Portlet Factory, we'll evaluate it against the product viability criterion and we'll evaluate IBM against the company viability criterion. Let's get on to the evaluation.

PORTLETS

As we mentioned above, portlets are the mechanisms for accessing data and applications from portals. You'll use portlets to facilitate your customers' access to customer, order, incident, and contract data, and their access to the services and functions of your applications. Clearly, portlets are critical technologies for supporting Customer Scenarios[®] in your customer portal. We'll evaluate them using these subcriteria:

- Packaged portlets
- Third-party portlets

- Portlet development tools
- Development samples and templates
- Portlet architecture
- Portlet-to-portlet communication
- Standards support for JSR 168 and WSRP

In Table A, we present our evaluation of WebSphere Portlet Factory 5.11.3 against these subcriteria. The information in the table gives you an idea of what WebSphere Portlet Factory adds to a portal technology platform portlet support. Following the table, we'll get into more detail about how you can use WebSphere Portlet Factory to speed portlet development.

Portlets	
Evaluation Criteria	Evaluation
Packaged Portlets	<p>WebSphere Portlet Factory does not include complete, pre-built portlets. Rather, it packages more finely granular, pre-built, and configurable portlet functionality or subservices in components called Builders. Builders are software automation components that capture and automate functionality, such as executing SQL queries, accessing Web Services, and creating a portlet's UI. WebSphere Portlet Factory packages 165 Builders. Builders are extensible. Your developers may modify the packaged Builders or create custom Builders.</p> <p>To create complete portlets, developers assemble the Builders needed to perform all the work of a portlet into portlet Models. WebSphere Portlet Factory generates complete, deployable portlets from the Builders in a Model. Portlets are generated at runtime when they're first requested and then reused throughout the duration of a session.</p> <p>IBM does offer pre-packaged, portal-based composite applications that are built using WebSphere Portlet Factory and are offered as separately-priced add-ons to WebSphere Portlet Factory. For example, IBM's Workplace Dashboard Framework includes a set of configurable dashboards for various roles and industries, such as sales, executive, and manufacturing.</p>
Third-Party Portlets	IBM Portlet Factory does not package third-party portlets. No third party offers Builders.
Portlet Development Tools	WebSphere Portlet Factory is a portlet development toolset. You assemble portlet Models from finely-grained components which you configure to customize their processing through their input and output parameters. For example, to connect to a database, you use the SQL Call Builder, which has input parameters that include the data source needed for the connection and the SQL to execute.

Portlets (continued)	
Evaluation Criteria	Evaluation
Portlet Development Tools (continued)	<p>You can specify Builders' input statically within the assembly process to generate single-purpose portlets for all of your customer portal users. Or, you can separate parameter specification from the assembly process to create general-purpose, customizable portlets that can be personalized for customer segments or even individual customers. Developers create WebSphere Portlet Factory components called Profiles for creating general-purpose, customizable, and personalizable portlets. A Profile contains value(s) for each of the parameters for the Builders in a Model that are used to specialize its functionality and/or audience. Developers can create and apply multiple Profiles to one Model, generating many highly-customized variations of a single portlet Model. For example, a developer could create an ordering portlet that queries a database and enables create, read, update, and delete of order data. The developer may choose to profile the data access Builders, the UI look and feel Builders, the access control Builders, the locale, and even the logic and process flow Builders. Developers could create a different Profile for each customer segment or even each customer.</p> <p>Profiling a Model is a task for developers. Developers, however, can enable end users to specialize, customize, and/or personalize Models by building browser-based wizards called Customizers. Customizers present only those parameters that you'd like an end user to specialize, insulating your end users from the complexity of Builder parameters and protecting your portal from portlets that don't use its resources efficiently. Customizers can be used at design time or at runtime. By providing your customers with easy-to-use Customizers, you can let them create or configure the portlets that they need to accomplish their Customer Scenarios[®].</p>
Development Samples and Templates	<p>WebSphere Portlet Factory packages more than 50 sample Models in its product distribution. These Models help developers learn how to use and apply many of the product's key Builders. For example, there are sample Models that use beans, SAP R/3 and BW, PeopleSoft, Siebel, Domino, databases, and Web Services. Many of these Models are also profiled so that developers can learn how to apply this technology.</p> <p>In addition, WebSphere Portlet Factory includes a set of "New Model" wizards, which are templates that streamline the Model creation process for common types of portlets such as database and Domino access.</p> <p>WebSphere Portlet Factory also packages Profile Set selection handlers. A profile selection handler contains the logic for how to map a Profile to a Model. The pre-packaged handlers do explicit Profile set selection or Profile set selection based on file segment, J2EE environment, LDAP, or locale. For example, the WebSphere Portal Group selection handler will select the profiles for a Model based upon what Portal Group within a WebSphere Portal the end user belongs to.</p>
Portlet Architecture	<p>You can configure WebSphere Portlet Factory to generate either IBM API portlets for the IBM WebSphere Portal or JSR 168 portlets.</p> <p>Organization. WebSphere Portlet Factory portlets are organized as Model 2 (MVC) applications.</p> <p>Infrastructure. Models generate portlets that deploy on a wide variety of J2EE application servers.</p> <p>Structure. Models generate Java portlets that have JSP UIs. J2EE facilities are used for database and application access.</p>

Portlets (continued)	
Evaluation Criterion	Evaluation
Portlet-to-Portlet Communication	Portlet-to-portlet communication is accomplished using the Event Declaration and Event Handler Builders, or the Shared Session Builder for portlets that need to communicate across pages. Portlets generated by WebSphere Portlet Factory can also participate in portlet-to-portlet communication on the IBM WebSphere Portal platform through the C2A (Click to Action) Menu Builder.
JSR 168	Portlets developed from WebSphere Portlet Factory Models can be automatically deployed as JSR 168-compliant portlets.
WSRP	JSR 168 portlets generated by WebSphere Portlet Factory can function as WSRP-compliant, remote portlets. As a portlet development tool, WSRP compliance is not directly applicable to WebSphere Portlet Factory. WSRP is a standard for portal platforms that lets users access portlets on remote portal platforms at runtime.

© 2006 Patricia Seybold Group

Table A. We present our evaluation of the portlet subcriteria for WebSphere Portlet Factory 5.11.3 in this table.

Our Take on WebSphere Portlet Factory Portlets

Portlet support in WebSphere Portlet Factory is very good. After all, the product is a portlet development tool and its component assembly and mass customization approach to portlet development can deliver significant productivity benefits to your development staff. WebSphere Portlet Factory is also strong in our evaluations for packaged portlets, samples and templates, portlet architecture, and portlet-to-portlet communication.

While the product can simplify and speed portlet development, your use of WebSphere Portlet Factory requires development skills. Its tools are visual and many of its interfaces are wizard-based, but they require knowledge of the structures and mechanisms of Web applications and portals. Business analysts may be able to use WebSphere Portlet Factory's Builders to create portlet UIs, but they would need developers to create the data access Models and any access to Java resources, for example.

WebSphere Portlet Factory does enable end users to configure and personalize portlets using browser-based wizards, called Customizers. However, developers have to build those Customizers. In our evaluations of portal technology platforms, we have cited the lack of development tools appropriate for

business users as a key limitation. We cite WebSphere Portlet Factory for that limitation, too.

PACKAGED PORTLETS AND THIRD-PARTY PORTLETS. IBM puts 165 Builders in the WebSphere Portlet Factory box. They package much of the functionality that you'll need in your customer portal. There are Builders for:

- Portlet UI development
- Portlet-to-portlet communication
- Database access
- Java development for portlet logic, using J2EE facilities for access to external Web resources
- Java Bean access
- Web Services integration
- Domino application and data access
- SAP application and data access
- PeopleSoft application and data access
- Siebel application and data access
- Excel import and export

For example, developers would use the Data Page Builder and associated pages modifier Builders to automatically generate the portlet's UI from a database schema associate. Developers can then fine tune the UI a number of different ways, including adding Builders to the Model such as Link, Button, Check Box, Image, and Text. Builders for Domino and SAP access are especially rich in functionality. SAP Builders access both SAP applications and the SAP Business Warehouse. The SAP Builders are:

- SAP View and Form
- SAP Function Call
- SAP Batch
- SAP Properties
- SAP Read Table
- SAP Transaction
- SAP BW Data
- SAP BW Column Headers
- SAP Help Values

Portal technology platforms also package this breadth and depth of functionality, but they do so in complete portlets that your developers can reuse and extend using Java programming environments. Portlets built with WebSphere Portlet Factory are more easily reused and extended than Java portlets. Why? Because IBM designed Builders to be assembled into portlet Models and enabled Models to be mass customized through Profiles. Builder functionality is more modular and more finely granular, increasing its reusability and facilitating its customization. Its interfaces are completely parameterized and can be represented in XML documents. Its implementation is completely exposed in its interfaces. No functionality is hidden.

Note that, while 165 packaged Builders deliver a lot of functionality, you might need additional capabilities in your portlets. No problem—WebSphere Portlet Factory gives you the tools and the sample structures to let your developers create custom Builders. So, while you can certainly link in Java code into an IBM Model, you get better reuse and easier maintenance by creating a custom Builder to add to your Model. So, rather than maintaining custom code in many places, you maintain custom components in a few places.

PORTLET DEVELOPMENT TOOLS. We've been endorsing application development via framework-based component and mass customization for 15 years. It's an approach that promotes reuse and facilitates refinement and adaptability. It's an approach that improves developer productivity and results in application functionality that more closely matches user requirements. We're happy to recommend WebSphere Portlet Factory as a toolset that lets you adopt this best practice.

WebSphere Portlet Factory can save you time in developing portlets and time in managing them. You'll need a few Java developers to design the overall architecture for your portal development, the Java resource access specifications, the data access specifications, and to create custom Builders. You can then have junior developers or developers new to Java create the portlets by assembling Builders in Models, and create the profiles. Once the Models have been created, you won't need the highest level of Java development skills to generate, refine, and manage them. Your best developers will be freed to focus on the most complex development tasks.

Props to IBM for Profiles and Customizers. These structures really innovate portlet development, deployment, and management. Your developers can profile all the input parameters of all WebSphere Portlet Factory's Builders. As a result, you can create a customizable portlet portfolio across the dimensions of functionality and users. For example, you can Profile portlets by database, database view, database table, or database column. Then, you can customize each of them by customer segment, customer role, customer organization, or even individual customer. One Model, many, many customizations. And, you don't have to account for all of the customization requirements at design time. Customizers enable your developers to let your customers and users change portlet functionality at runtime based on who they are and the work that they want to do.

Patty Seybold has recommended that you should create thousands of customer portals to make it easy for your customers to do business with you by letting them help themselves to do their work. This writer recognized that recommendation as ideal, but was critical of that recommendation for the development and management work that it would involve for your developers and administrators. WebSphere

Portlet Factory lets you achieve Patty's vision with a single portlet Model and a few portlet Profiles. WebSphere Portlet Factory eliminates this writer's criticism.

DEVELOPMENT SAMPLES AND TEMPLATES. As we show in Table A, WebSphere Portlet Factory has good samples and examples of Models and Profiles. They're part of the product distribution and they're automatically installed with the product.

PORTLET ARCHITECTURE. Portlets are Web applications. So their architecture must support a user interface, application logic, and access to persistent resources such as data or external applications. Internally, portlets must have the mechanisms needed to integrate within their deployment environments—the portals through which they're accessed.

WebSphere Portlet Factory generates portlets with a good portlet architecture. WebSphere Portlet Factory portlets are Java portlets and leverage J2EE facilities for accessing external resources: JNDI and JDBC for database access, WSDL, SOAP, and HTTP for Web Services access, JAAS and JCO for application access, JMS for application messaging, for example. Those of you with .NET infrastructures will have to find another way to improve development of your portlets, although you can access remote WebSphere Portlet Factory portlets via WSRP or create portlets using Web Services created in .NET.

WebSphere Portlet Factory portlets' Model 2 organization is the current best practice in application design patterns. Model, view, and controller components are all implemented with Java technologies. In WebSphere Portlet Factory release 6, IBM plans to deliver Model 2 support through implementation of the Java Server Faces (JSF) specification.

PORTLET-TO-PORTLET COMMUNICATION. Portlet-to-portlet communication is very good. WebSphere Portlet Factory portlets can communicate with each other, and with JSR 168 and/or IBM API portlets through events. WebSphere Portlet Factory provides comprehensive event support in its Builders. It has a set of predefined events and the flexibility for your developers to specify custom events. The WebSphere Portlet Factory Designer has Builders for declaring which events should be handled within a portlet and for specifying handlers for

your events. IBM's C2A (Click to Action) events are supported, too.

These are the predefined events:

- OnWebAppLoad is triggered each time a new instance of the Model's WebApp is instantiated for a session.
- OnRequest is triggered each time an external request is made to the WebApp. External requests are typically made from Web browsers.
- OnUnhandledError is triggered when an error, for which no error handler has been specified, occurs.
- *pageName*:OnPageLoad is triggered each time a specified page is requested. Each page in a Model may have its own OnPageLoad event declaration.

Your developers can specify custom events with the Event Declaration HTML Event Action Builder. Custom events are based on user interactions with HTML controls at the browser or on back-end application processing.

JSR 168 AND WSRP. Developers can configure and run portlet Models to generate as JSR 168-compliant portlets.

PRODUCT VIABILITY

You want to purchase a portal technology platform that is well proven and widely used for your type of business. You also want a product that you can implement within a budget and schedule, and a product that will continue to be able to address your requirements in future versions. In other words, you want a viable product.

We've identified six subcriteria in evaluating product viability. We feel that they apply to any type of software product. The product viability subcriteria are:

- Product background
- Installed base
- Target market(s)
- Pricing
- Product plans
- Competition

In Table B, we evaluate these subcriteria for WebSphere Portlet Factory 5.11.3. Our analysis fol-

lows.

Product Viability			
Evaluation Criteria	Evaluation		
Product Background	Version	Date	Key Features
	5.11	December 2005	<ul style="list-style-type: none"> Project creation enhancements Enhanced error checking for File Upload Builder
	5.10	June 2005	<ul style="list-style-type: none"> Data Services Builders for portlets built on SOA Excel Import Builder to generate a portlet that displays Excel data Rich Data Definition Builder allows additional metadata for portlet schemas for functions such as formatting, validation, and name value pair translations
	5.9	November 2004	<ul style="list-style-type: none"> Siebel Extension: graphical tools for automating, developing, customizing, and maintaining portlets that access Siebel applications and data SAP Builder for portlets that access SAP Business Warehouse via XMLA
	5.8.2	June 2004	<ul style="list-style-type: none"> PeopleSoft Extension: graphical tools for automating, developing, customizing, and maintaining portlets that access PeopleSoft applications and data
	5.8	March 2004	<ul style="list-style-type: none"> JSR 168 compliance
	5.7	November 2003	<ul style="list-style-type: none"> Ready for IBM WebSphere Studio software validation SAP Extension: graphical tools for automating, developing, customizing, and maintaining portlets that access SAP applications and data Support for JBoss 3.2.1 (Java) Bean Master Detail Builder
	5.6.2	August 2003	<ul style="list-style-type: none"> Support for IBM WebSphere 5.0
	5.5	December 2002	<ul style="list-style-type: none"> Portlet Wizard Builders for database, JMS, and page automation Enhancements for producing and consuming Web Services
	5.0	October 2002	Introduction

Product Viability (continued)	
Evaluation Criteria	Evaluation
Installed Base	IBM claims that WebSphere Portlet Factory has an installed base of more than 100 customers in the Global 2000. Recognizable customer names in this customer segment include: Cisco, DuPont Fidelity Information Services, and Northwestern Mutual. WebSphere Portlet Factory also has an installed base of smaller organizations. We'd estimate the size of this installed base at approximately 100. So the total installed based is around 200 companies.
Target Market(s)	For WebSphere Portlet Factory, IBM targets companies and organizations of all sizes in all industry segments, the target market for all portal technology platforms.
Pricing	The pricing model for WebSphere Portlet Factory is a per-seat charge of \$5,000 per seat for the Designer toolset and a per-CPU charge of \$20,000 per CPU for Profiling. IBM told us that its typical deal price for software ranges from \$50,000 to \$100,000, and that its customers typically spend an additional 50 percent for consulting services to build and deploy a complete portal solution.
Product Plans	IBM plans to introduce two new versions of IBM WebSphere Portlet Factory by mid-2006. These new releases will include native support WebSphere Portal 6.0, enhanced SOA architecture, and support for asynchronous portlets via AJAX.
Competition	WebSphere Portlet Factory competes most closely with the portlet development tools packaged in portal technology platforms.

© 2006 Patricia Seybold Group

Table B. This table shows how WebSphere Portlet Factory addresses our subcriteria for product viability.

Our Take on WebSphere Portlet Factory

5.11.3 Product Viability

WebSphere Portlet Factory is a three-year-old product with a technology heritage that dates back two years prior to its introduction. From functional and architectural perspectives, Bowstreet had taken it through multiple major versions, although the numerical designations of those versions appear as “point releases.” Combine this background with a reasonably-sized installed base, and you get a viable product.

Until this year, most customers used WebSphere Portlet Factory to build internal portals. In 2005, a number of the product's largest customers began to use Portlet Factory to accelerate the development and implementation of their customer portals.

Bowstreet Portlet Factory had always been very closely tied to IBM WebSphere Portal. The product

was introduced as a development enhancement to that platform.

COMPETITION. You're going to have portlets, lots of portlets, in your customer portals because your customer portal should contain all of the key Customer Scenarios[®] for each of your customer segments. J2EE portal technology platforms let you develop them from scratch or let you extend the packaged and third-party portlets that they package. Either way, you'll use a combination of the portlet development tools that are packaged with portal technology platforms and Java IDEs, making portlet development very similar to conventional Java development.

WebSphere Portlet Factory offers an alternative portlet development approach—portlet assembly with mass customization—that delivers increasing efficiencies as you increase the number of portlets in your customer portal. On one hand, no portlet devel-

opment tools compete directly with WebSphere Portlet Factory because no portlet development tools support the assembly and customization approach. But, on the other hand, WebSphere Portlet Factory does compete for your development budget. Your choice is conventional development or assembly and customization.

The acquisition of Bowstreet by IBM changes competition for WebSphere Portlet Factory. Bowstreet was an independent portal software supplier. IBM, of course, offers the IBM WebSphere Portal, and IBM competes fiercely with portal suppliers like Oracle that had been Bowstreet's partners. Going forward, WebSphere Portlet Factory will be tied closely to IBM WebSphere Portal. Developers will use it to build IBM API and JSR168 portlets.

COMPANY VIABILITY

You want to purchase a viable product from a viable company. A viable company is a going concern with increasing revenue, profits, numbers of customers, and (sometimes) numbers of products.

We've identified four subcriteria for evaluating company viability. They are:

- Company background (including its current status)
- Product lines (products and lines outside customer service)
- Customer base (for all products)
- Financials

Before the IBM acquisition, WebSphere Portlet Factory had been offered by Bowstreet, Inc., a small and young software supplier. Company viability was an issue in your selection of the product. That's no longer the case. There are no company viability issues with IBM. IBM is one of the oldest, the largest, and the most successful software, hardware, and services companies in the industry.

THE BOTTOM LINE

WebSphere Portlet Factory provides a comprehensive set of facilities for developing, deploying, and managing the portlets in your IBM WebSphere Portal customer portal. Its component assembly and mass customization approach to portlet development can speed and simplify the implementation of your custom portal.

We recommend that you add WebSphere Portlet Factory to the toolkit of your portal platform. It's a viable product from an established company.

Contact Info:

IBM Corporation
 4 Technology Park Drive
 Westford, MA 01886
 Dee Zepf
 Product Manager, WebSphere Portlet Factory
 Phone: 978.399.7619
 Email: dee@us.ibm.com
 Internet: www.ibm.com