



PORTAL AGGREGATION AND INTEGRATION LAB

Interoperability between Microsoft Office SharePoint Server 2007, BEA WebLogic Portal, and IBM WebSphere Portal (with Mainsoft)

Published: August 2008

Abstract

Enterprises rely on portals to facilitate information-sharing and speed business processes. Portals provide a convenient, centralized way for workers to gain insight into the business and to collaborate on projects through a single point of entry, in the form of a Web-based user interface. In this lab, Microsoft demonstrated integration between Microsoft® Office SharePoint® Server 2007 and BEA WebLogic Portal and IBM WebSphere Portal software solutions. In this scenario, Office SharePoint Server 2007 is the data provider, that is, Office SharePoint Server 2007 exposes its data to enable users to consume data from within third-party portal user interfaces. This need may arise, for example, when an organization has an enterprise portal infrastructure in which it needs to aggregate and reuse existing Office SharePoint Server 2007 user interface elements from a departmental collaboration portal.

Using the solutions detailed in this document, organizations can achieve integration between enterprise-level and departmental-level portals for a seamless, efficiency-enhancing employee experience.

This is a preliminary document and may be changed substantially prior to final commercial release of the software described herein.

Table of Contents

PORTAL AGGREGATION AND INTEGRATION LAB.....	1
Interoperability between Microsoft Office SharePoint Server 2007, BEA WebLogic Portal, and IBM WebSphere Portal (with Mainsoft).....	1
Abstract.....	1
Introduction.....	4
Lab Scenarios	6
Interoperability between Microsoft Office SharePoint Server 2007 and IBM WebSphere Portal with the Mainsoft SharePoint Federator.....	7
Runtime Environment Scenario	8
Sharepoint Federator authentication methods	9
Creating a Portlet	10
Configuring the portlet.....	12
Adding/configuring a second portlet.....	15
More information on the Sharepoint Federator.....	17
Design-Time Environment Scenario.....	18
Interoperability between Microsoft Office SharePoint Server 2007 and BEA WebLogic Portal with BEA AquaLogic .NET Application Accelerator	22
Build a Microsoft ASP.NET page that exposes a SharePoint list.....	23
Extracting and exposing data from SharePoint site pages.....	24
Configuring the WSRP Producer	29
WSRP Producer configuration.....	29
Enabling UNT Authentication.....	29
Defining Credential Mapping.....	29

Hardcoding Credentials, an alternative to WSRP Producer Configuration 29

Adding the new pages to the list of ASP.NET pages that can be exposed 31

Creating new remote portlets on BEA WebLogic Portal 33

Appendix: The Interop Vendor Alliance..... 36

Related Links 37

Introduction

Microsoft® Office SharePoint® Server 2007 is a suite of content management and collaboration capabilities that helps organizations improve business processes by consolidating intranet, extranet, and Web applications within one integrated platform.

Although Office SharePoint Server 2007 provides easy-to-use functionality for organizations to create, approve, and publish Web content to enterprise portals or departmental workgroup sites or collaboration portals, the enterprise domains of some organizations that have adopted Office SharePoint Server 2007 can also include other portal technologies. Key business data can be available to users across a range of applications. For instance, in heterogeneous portal environments, the companywide intranet used by all employees may be delivered by, among others, a BEA-, IBM-, or SAP-based portal solution, while at the departmental level Office SharePoint Server 2007 may be used instead. Integration between different technologies is needed in order to avoid the creation of information silos. Solutions are required to bridge the differences between varied technologies, enabling a two-way exchange of information that allows organizations to reuse data among portals.

This interoperability lab focused on reusing user interface portal components (Web Parts), such as document lists, task lists, calendar data, and more, between Office SharePoint Server 2007 and Java-based portals. As shown in Figure 1, In this scenario Office SharePoint Server 2007 is the data provider, that is, Office SharePoint Server 2007 exposes its data and enables end-users to consume data from the user interfaces of Java-based portals.

For the complementary scenarios where Office SharePoint Server 2007 needs to aggregate data from various sources, users can rely on technologies such as Web services; the Business Data Catalog feature of Office SharePoint Server 2007, which provides an easy way to integrate business data from back-end server applications; and database connections to consolidate data on a single portal in a way that makes it easier for users to consume. This scenario might be explored in future labs.

In this lab we tested two scenarios:

1. Office SharePoint Server 2007 as data provider using the Mainsoft SharePoint Federator for WebSphere Portal
2. Office SharePoint Server 2007 as data provider using the BEA AquaLogic .NET Application Accelerator

Lab Scenarios

In cooperation with other Interop Vendor Alliance (IVA) members and partners, Microsoft tested the integration capabilities between Office SharePoint Server 2007 and Java-based portals in real-world business scenarios at the Portal Aggregation and Integration Lab. In this scenario, we tested two integration products to show how they can be used to consume SharePoint Web Parts from a Java-based portal: the Mainsoft SharePoint Federator for WebSphere Portal and the BEA AquaLogic .NET Application Accelerator for BEA WebLogic/AquaLogic Portals. The solutions tested demonstrated the interoperability between Microsoft Office SharePoint Server 2007 and portal technologies from BEA and IBM.

Portal Aggregation and Integration Lab

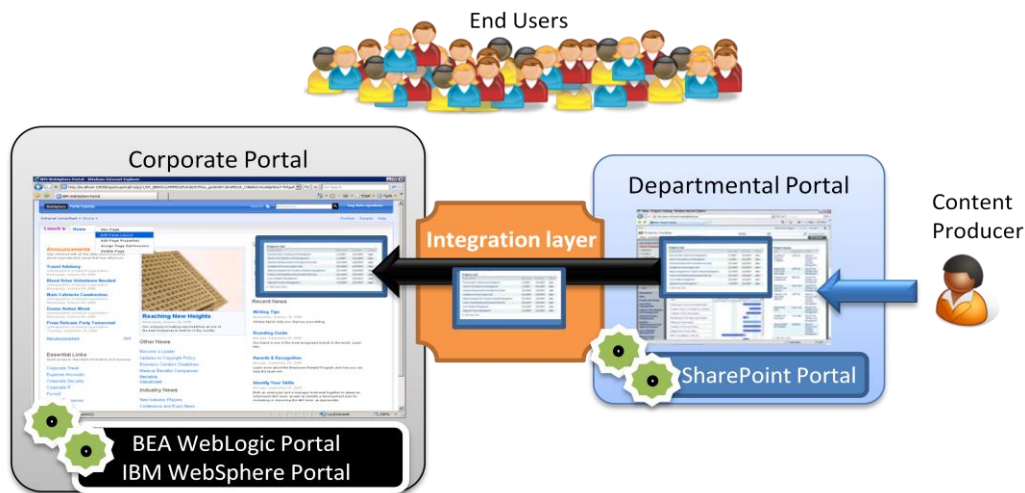


Figure 1. Interoperability between Microsoft Office SharePoint Server 2007, BEA WebLogic Portal, and IBM WebSphere Portal

Interoperability between Microsoft Office SharePoint Server 2007 and IBM WebSphere Portal with the Mainsoft SharePoint Federator

The Mainsoft SharePoint Federator is a solution that helps organizations using IBM WebSphere Portal software as their enterprise portal solution to aggregate contents from departmental Office SharePoint Server 2007 sites into the WebSphere environment. Using the SharePoint Federator, organizations that also have a significant investment in Microsoft .NET assets, such as Web content in Office SharePoint Server 2007, can easily reuse these assets within the WebSphere enterprise portal environment.

This lab demonstrates use of the SharePoint Federator to expose SharePoint Web Parts, such as announcements, document libraries, contact lists, and more, within an IBM WebSphere portal.

There are two components to this lab: a runtime environment and a design-time environment. In the runtime scenario, we will show how the SharePoint Federator List Viewer portlet can be used to display document libraries or other list views, such as announcements.

The runtime environment requires installing a component on the IBM WebSphere Portal Server that will enable the portal to consume basic information from Office SharePoint Server 2007. An IT administrator can perform the installation process; no development experience is necessary.

The second component involves displaying more complex Web Parts that might require customization. The SharePoint Federator ships with Mainsoft Federator Portlets, which includes the source code of List Viewer and Detail Viewer portlets so that developers can customize them to meet different requirements. From the code, a developer can also see how the SharePoint Federator software development kit (SDK) can be used to build portal components capable of displaying content in Office SharePoint Server 2007 from within an IBM WebSphere portal. Using the Mainsoft SDK, developers can also customize portal functions. For instance, portal user authentication and authorization can be integrated with an organization's existing identity management system. Customization to meet heightened security requirements is also possible.

Runtime Environment Scenario

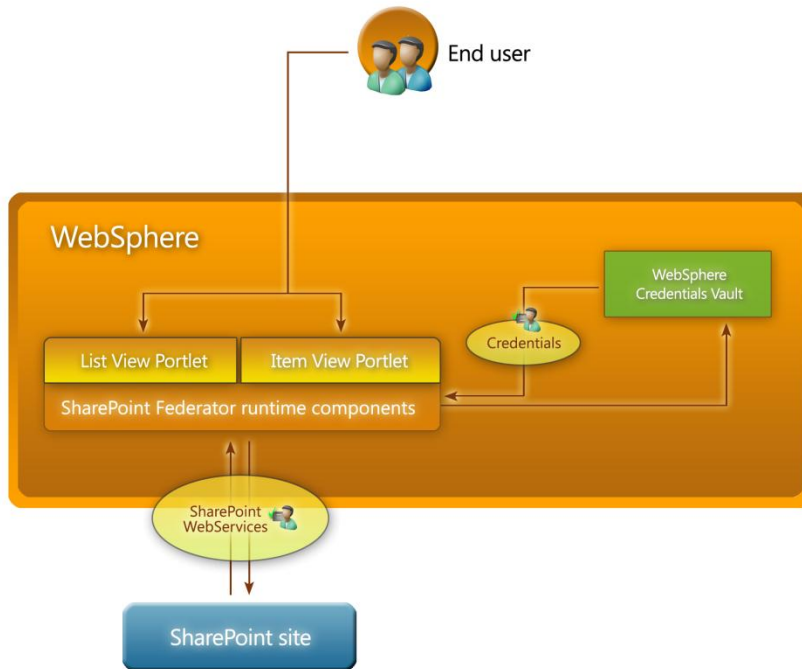


Figure 2. List Viewer portlet and Item Viewer portlet appearing on WebSphere portal, using SharePoint-based Web services and the WebSphere Credential Vault

For this scenario, we first installed the Mainsoft SharePoint Federator runtime components through the Portlet Management options in the WebSphere portal.

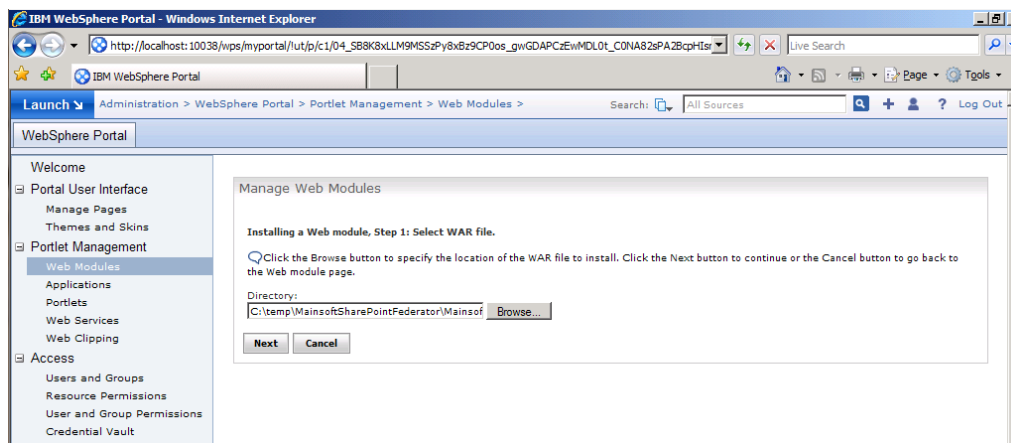


Figure 3. Installing runtime components

Sharepoint Federator authentication methods

The SharePoint Federator supports several different user authentication methods. The first and most preferable of these, due to ease of use, is to create a user name and password using the WebSphere Credential Vault, as shown in Figure 4. A user name and password may also be entered when configuring the List Viewer portlet, as shown in Figure 9.

In this lab, we used the WebSphere Credential Vault for user authentication. To enable user access to the SharePoint site, we registered the user in the WebSphere Credential Vault.

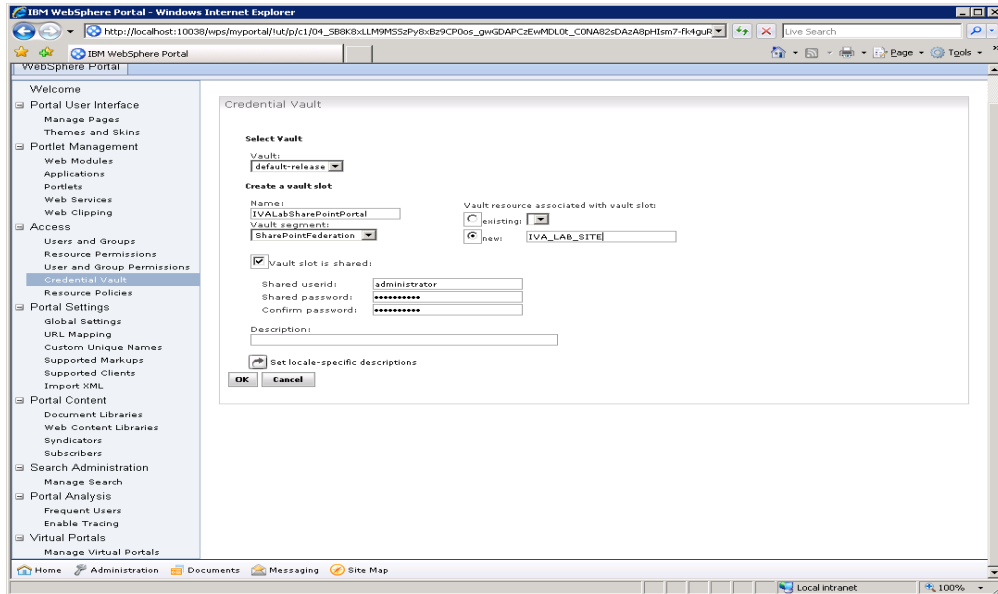


Figure 4. WebSphere Credential Vault

Creating a Portlet

Next, we went to the WebSphere portal home page.

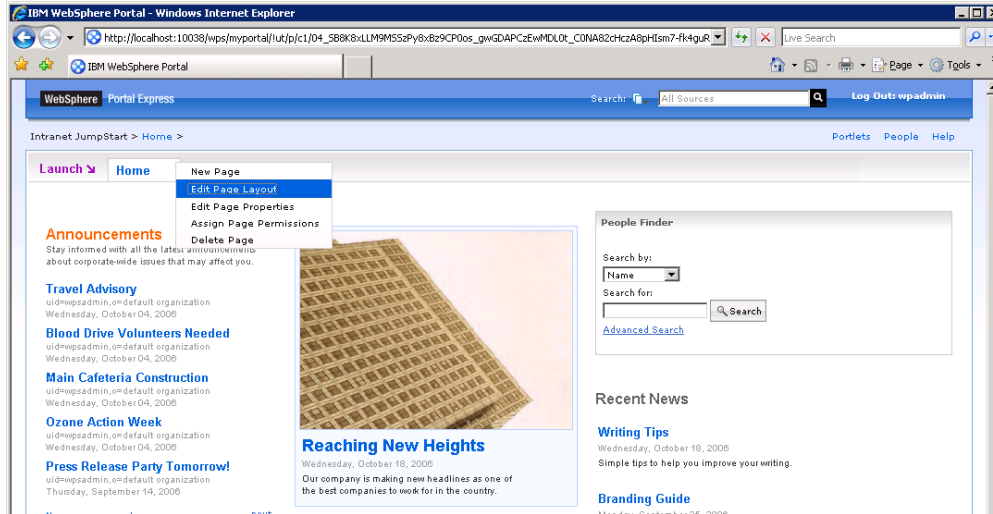


Figure 5. WebSphere portal home page

From the home page, we logged on with the user credentials and we went to the Edit Layout page where we can add a new portlet and position it on the page.

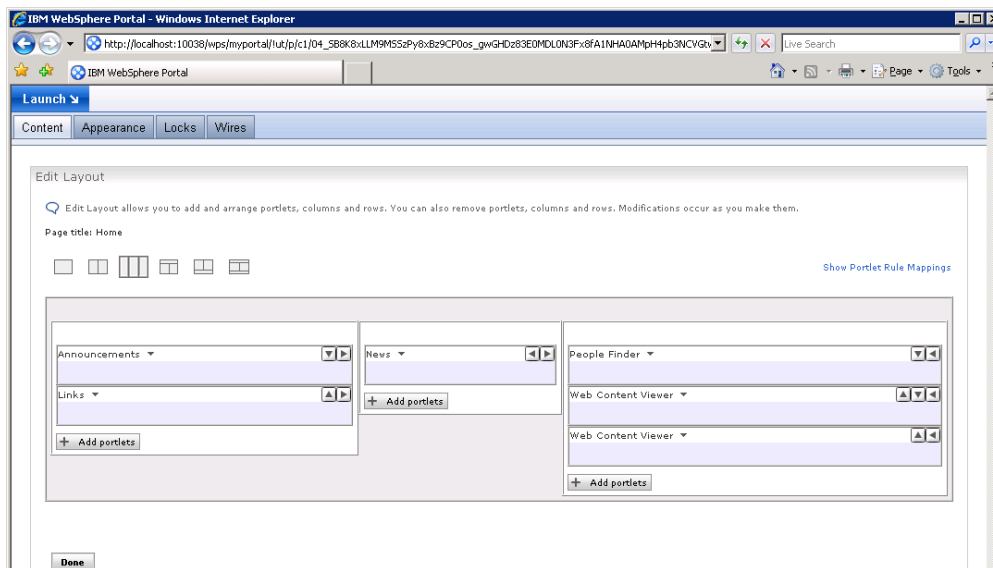


Figure 6. Edit Layout page in WebSphere portal

Mainsoft SharePoint Federator portlets include the List Viewer portlet, which provides secured access to SharePoint document libraries and lists, including contacts, announcements, tasks, events, and custom lists. From the list of portlets installed on the WebSphere site, we selected the SharePoint List Viewer.

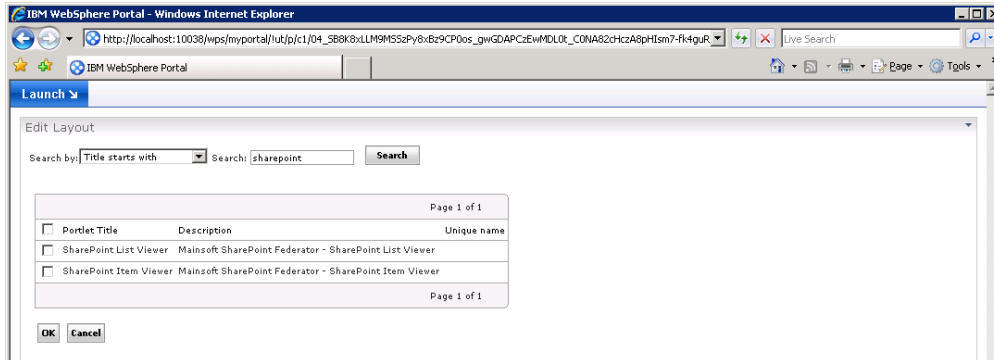


Figure 7. Selecting SharePoint List Viewer

The new portlet was added to the central column.

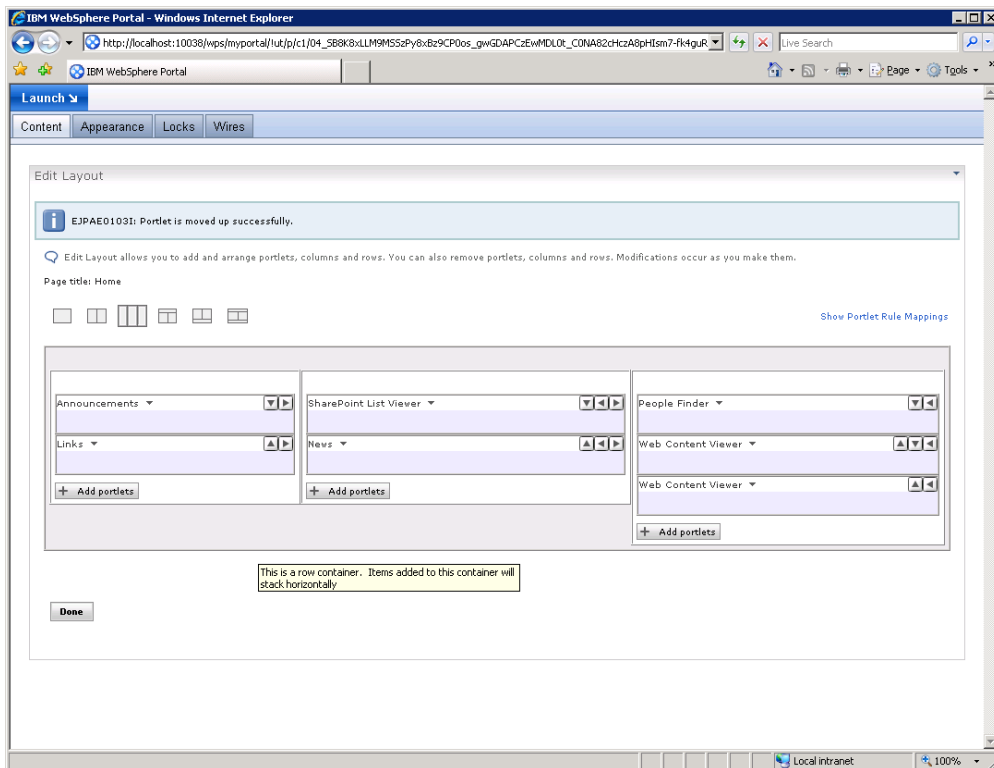


Figure 8. SharePoint List Viewer portlet added

Configuring the portlet

After exiting the Edit Layout page, we need to configure the List Viewer portlet to connect to the correct SharePoint site. We also need to configure other settings, including authentication, properties to display, and optionally other more advanced features, such as cascading style sheets (CSS), to be used in the portlet.

To connect to and display content from the SharePoint site document library, we provided the URL for the library, as shown in below.

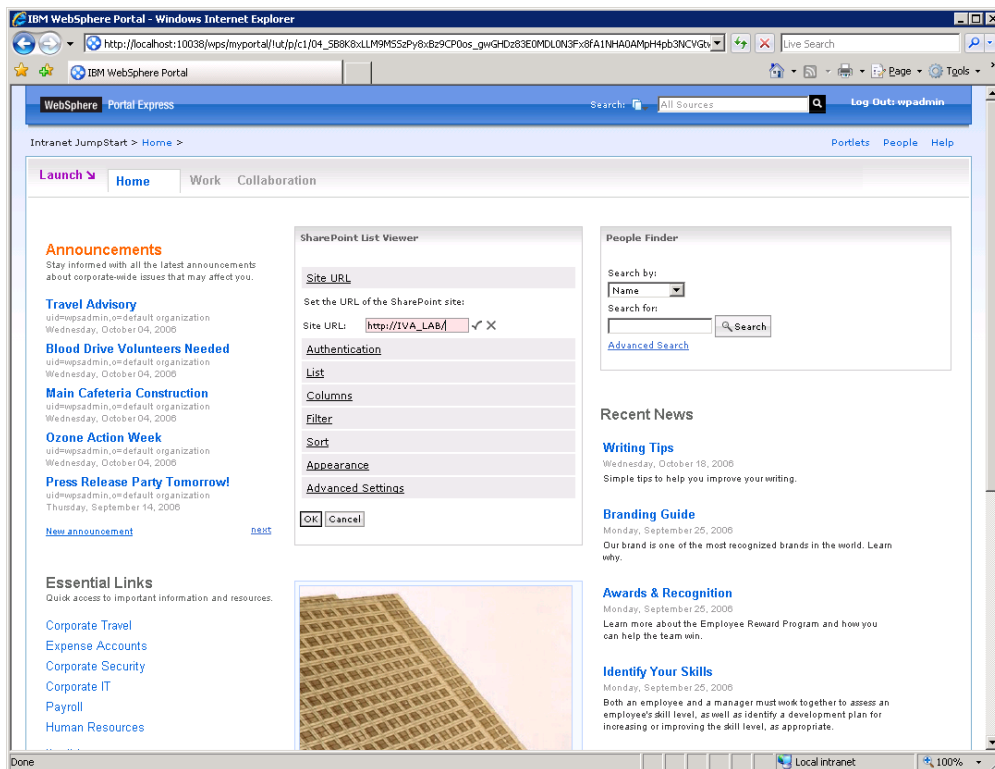


Figure 9. Entering URL for document library and user credentials for authentication

The List Viewer portlet offers a wide range of settings for customizing the appearance of the SharePoint list on the WebSphere portal, such as which columns are displayed within the portlet. The images below show how we designated which columns to display and formatting of columns.

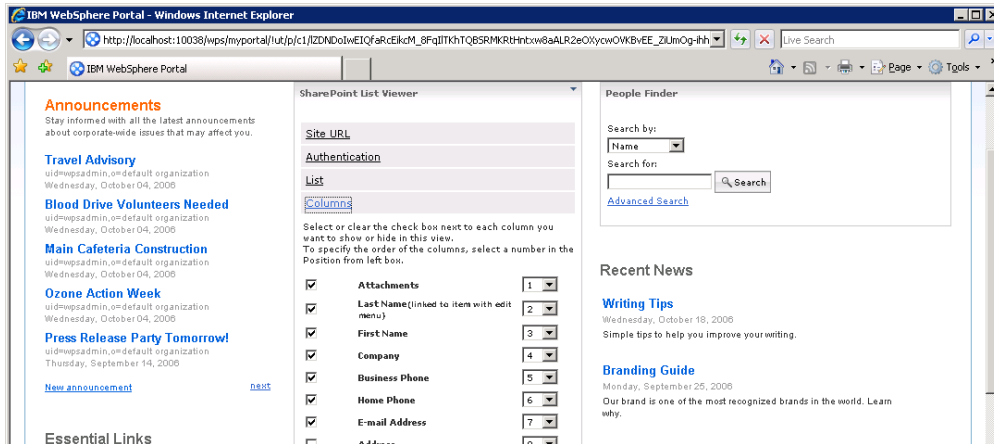


Figure 10. Designating which SharePoint list columns to display in the WebSphere portal

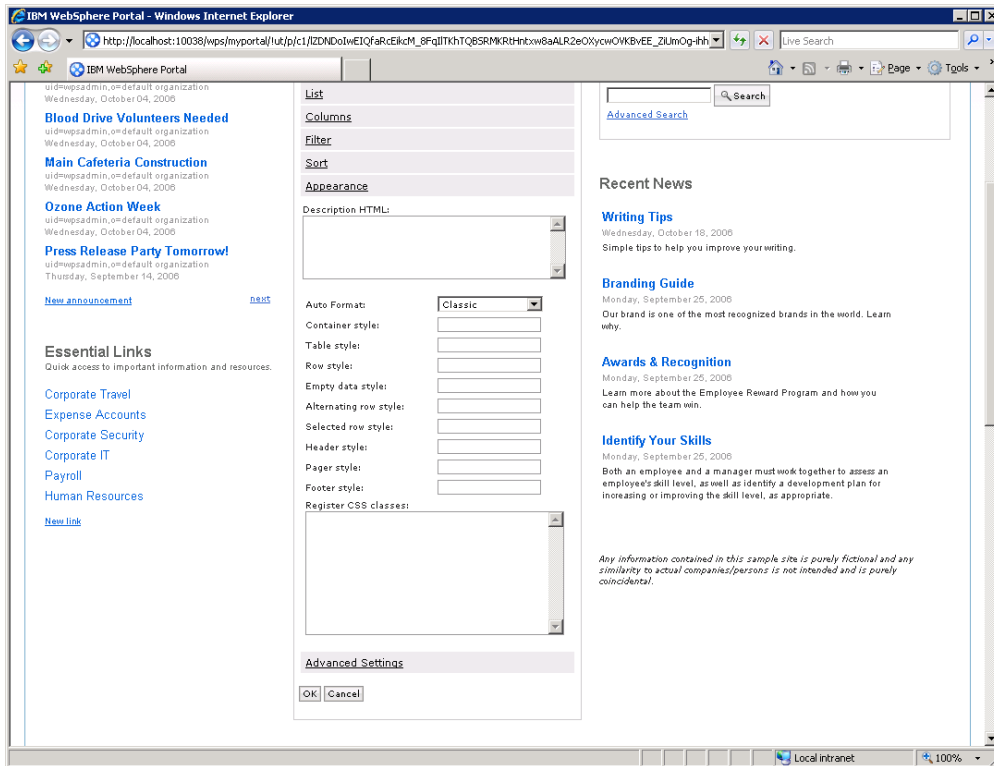


Figure 11. Choosing CSS formatting and styles for all List Viewer modes

After we configured the portlet, the WebSphere home page displayed the list of documents in the SharePoint document library and the columns of document details.

Figure 12 and Figure 13 show a comparison of the document library on the SharePoint site, and as it appears on the WebSphere portal.

This system library was created by the Publishing feature to store documents that are used on pages in this site.




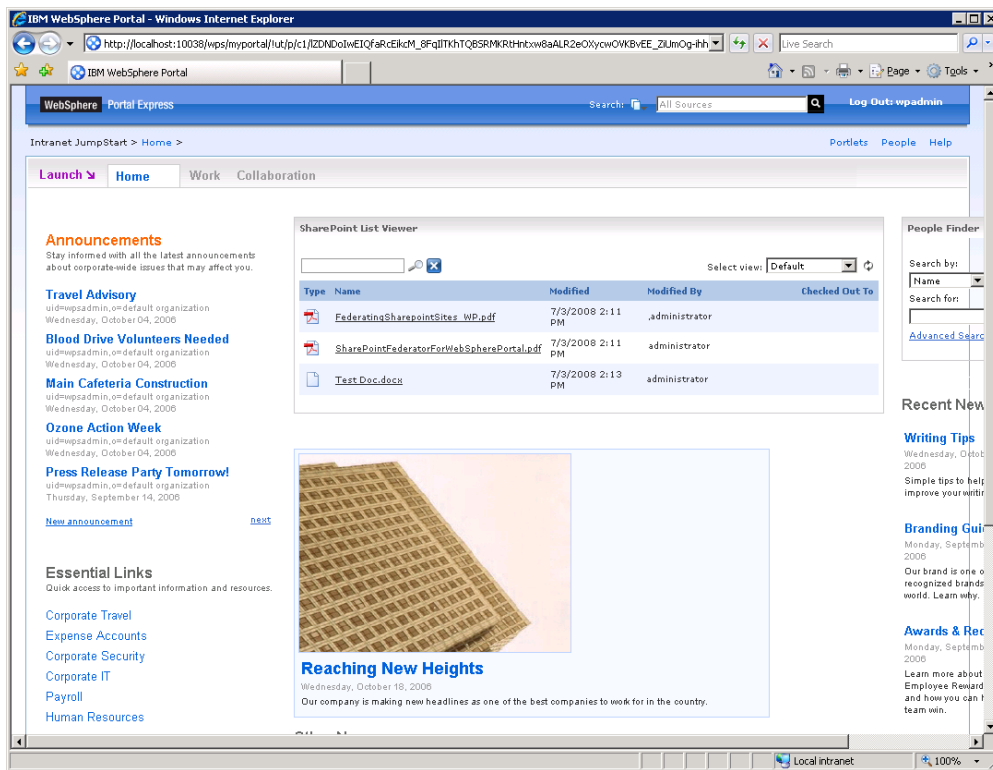



Type	Name	Modified	Modified By	Checked Out To
	FederatingSharepointSites_WP	7/21/2008 9:37 PM	WS03R2_SQL05\administrator	
	SharePointFedoratorForWebSpherePortal	7/3/2008 2:11 PM	WS03R2_SQL05\administrator	
	Test Doc	7/3/2008 2:13 PM	WS03R2_SQL05\administrator	

Figure 12. Document Library as it appears on the SharePoint site



The screenshot shows the IBM WebSphere Portal home page. A central portlet titled "SharePoint List Viewer" displays a table of documents from a SharePoint site. The table has columns for Type, Name, Modified, Modified By, and Checked Out To. The documents listed are:

Type	Name	Modified	Modified By	Checked Out To
	FederatingSharepointSites_WP.pdf	7/3/2008 2:11 PM	administrator	
	SharePointFedoratorForWebSpherePortal.pdf	7/3/2008 2:11 PM	administrator	
	Test Doc.docx	7/3/2008 2:13 PM	administrator	

The portal also features various other sections: "Announcements" with links to travel advisories and construction news; "Essential Links" for corporate travel, expense accounts, and security; and a "Reaching New Heights" article with an image of a building.

Figure 13. WebSphere portal home page showing the document library from the SharePoint site, with document names and details

Adding/configuring a second portlet

The Mainsoft SharePoint Federator also includes an Item Viewer portlet that displays the content and properties of a single item in the SharePoint list. Following the same steps used for the List Viewer portlet, we added an Item Viewer portlet.

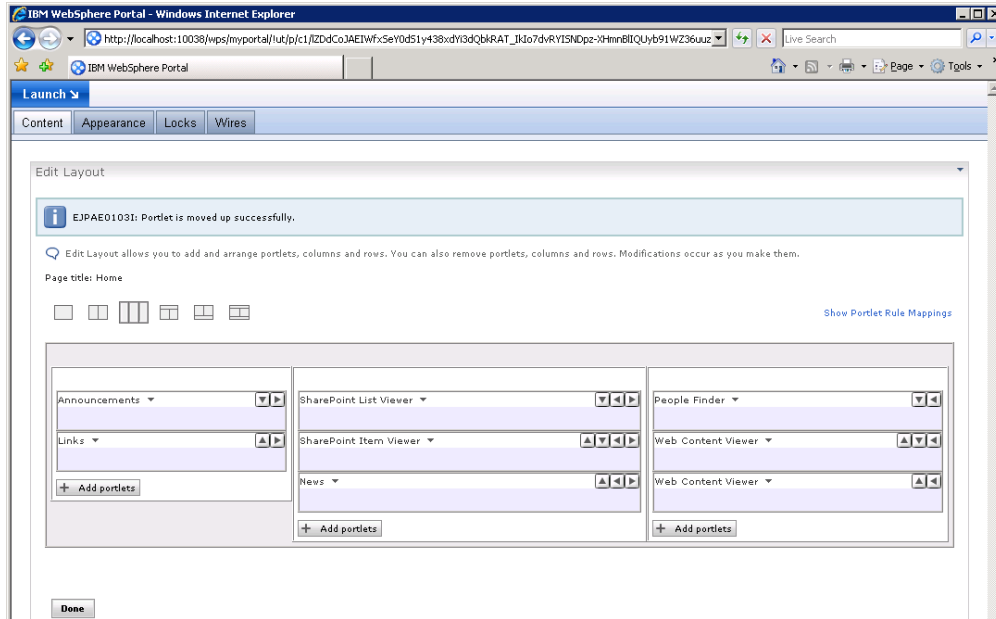


Figure 14. Adding the Item Viewer portlet

Next, we linked the Item Viewer portlet to the List Viewer portlet so that when an individual item is selected, the corresponding properties are displayed. As shown in Figure 14, to do this, we went to the Wires section of the page layout menu and using the cross-portlet integration capabilities in WebSphere Portal, we created a link between the List Viewer and Item Viewer portlets.

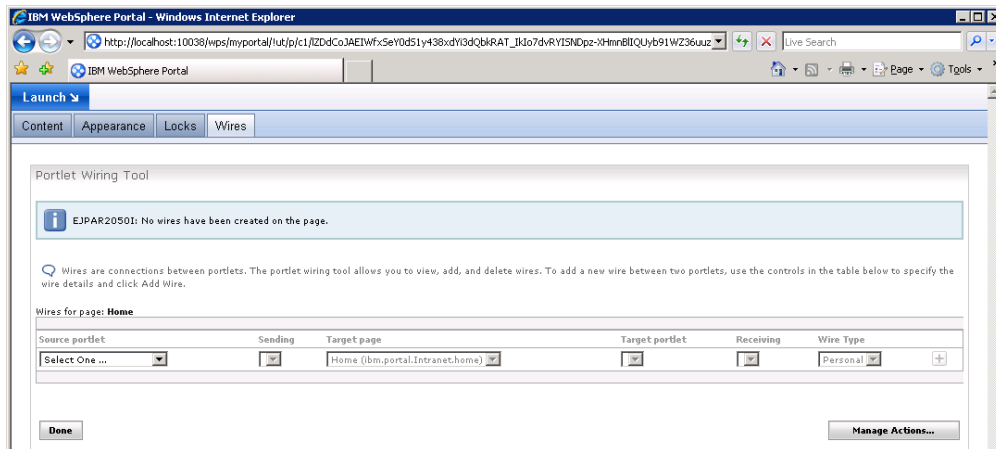


Figure 15. Linking the Item Viewer portlet to the List Viewer portlet

On the Wires section of the page, as seen below, we selected the source portlet and target portlet and specified the data that needs to be exchanged. In this case, only one parameter is required: ItemDetails.

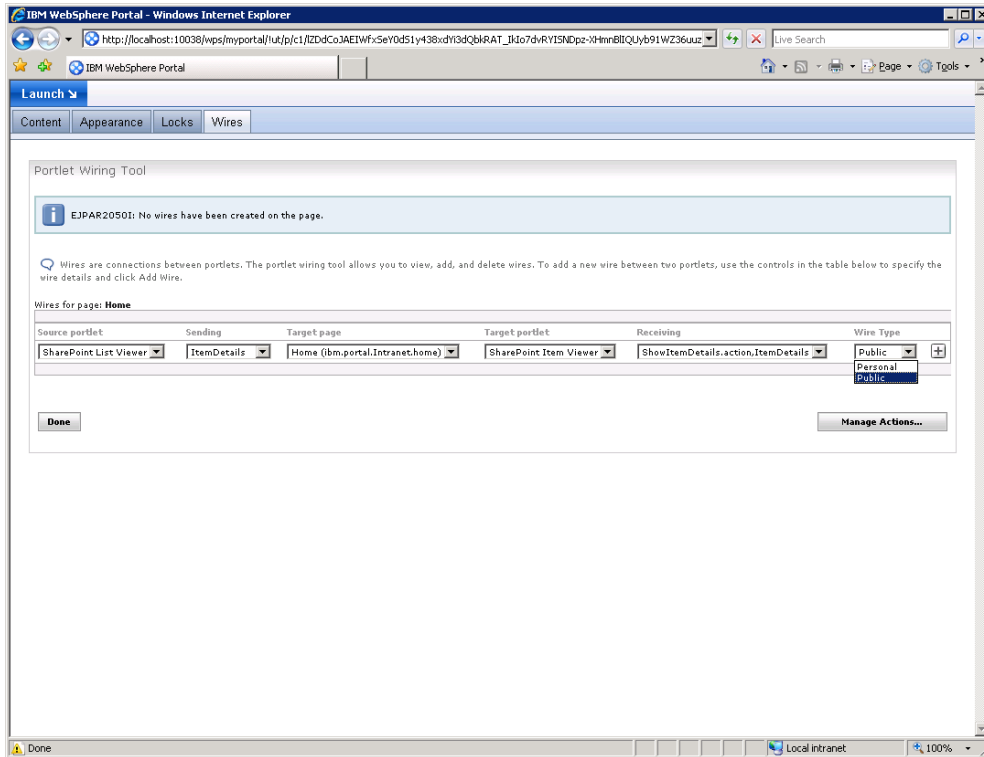


Figure 16. Designating source portlet, target portlet, and data to be exchanged

Now, when we select one item in the List Viewer portlet, the corresponding properties are loaded in the Item Viewer portlet as shown in Figure 16. Through the Item Viewer portlet, we can conduct more complex operations, such as checking out the document.

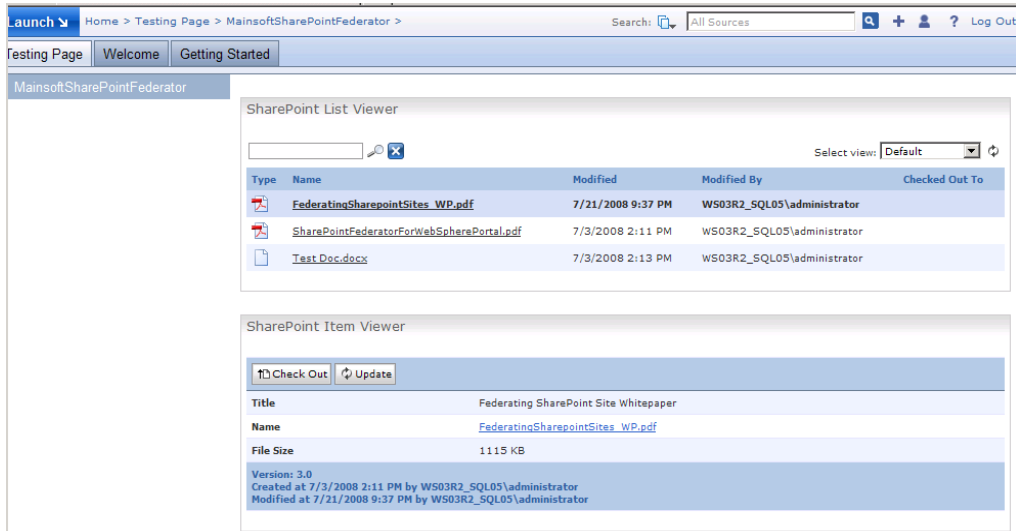


Figure 17. List Viewer portlet with corresponding properties in the Item Viewer

More information on the Sharepoint Federator

More details and information on the SharePoint Federator can be found in the following whitepaper:

http://www.mainsoft.com/solutions/pdfs/FederatingSharepointSites_WP.pdf

Design-Time Environment Scenario

The Mainsoft SharePoint Federator SDK provides a development environment to integrate Office SharePoint Server 2007 with IBM WebSphere Application and WebSphere Portal products.

The SDK includes a set of classes that can be used to connect a SharePoint site to an IBM WebSphere portal. These are shown in Figure 18.

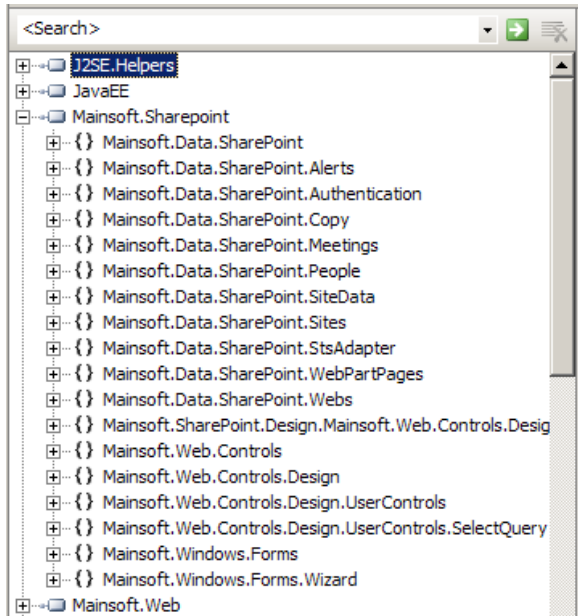


Figure 18. Classes exposed by SharePoint Federator

Classes included in the SharePoint Federator SDK provide full control over all components of a SharePoint site, as shown in Figure 19, greatly simplifying the development of integrated solutions. The SDK includes a SharePointDataSource control that facilitates binding operations on SharePoint lists, and a wizard that can be used at design time to configure all the parameters needed for a SharePoint connection. The SharePointDataSource control also enables the developer to filter, sort, and select what properties should be retrieved.

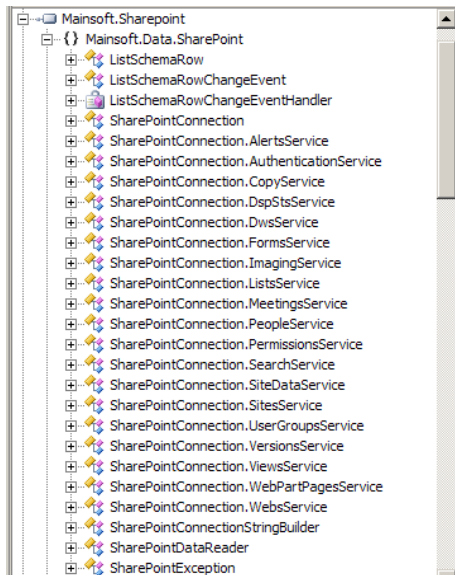


Figure 19. Classes available in the Mainsoft SDK to control access to a SharePoint site

The SDK also provides a class (SharePointConnection) that can be used to manage the connections to SharePoint sites similarly to the connection strings used to connect to databases.

To aid the development of solutions for IBM WebSphere Application and WebSphere Portal servers, the SharePoint Federator also includes a set of classes that can be used to manage other aspects, such as security settings and user profile information, as shown in Figure 20. There are classes to access the WebSphere Credential Vault that are used to map WebSphere users with users that have access to the SharePoint site. For instance, the SaveCredentials control can be used to capture credentials for SharePoint and store them in the Credential Vault.

Full control over security and user profile information gives customers maximum flexibility for building solutions that can be deployed in environments where security and authentication are managed by Windows Active Directory Services, as well as other technologies.

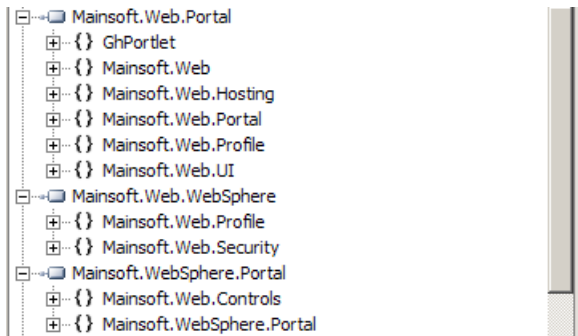


Figure 20. List of classes included in the SDK to control aspects such as Security, User Profile, and hosting environment

The SharePoint Federator SDK also includes a sample solution for the Microsoft Visual Studio® 2005 development system. It provides the complete source code for SharePointListView and SharePointItemView, giving the developer full control over these components. Using Visual Studio 2005, the developer can add portlets and change properties in an easy-to-use what-you-see-is-what-you-get (WYSIWYG) environment.

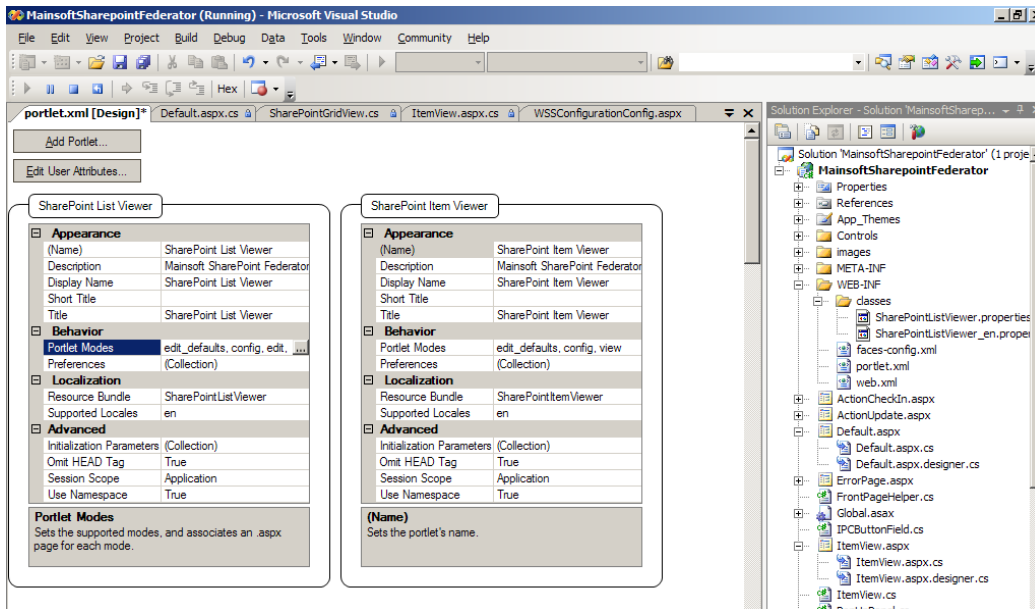


Figure 21. WYSIWYG editor for managing portlets

The SharePoint Federator also offers fully integrated debugging through which Visual Studio 2005 has access to the WebSphere debugging services. This is shown in Figure 22 and **Error! Reference source not found..** Developers can set breakpoints in Visual Studio 2005 and debug the control while it runs on the WebSphere site.

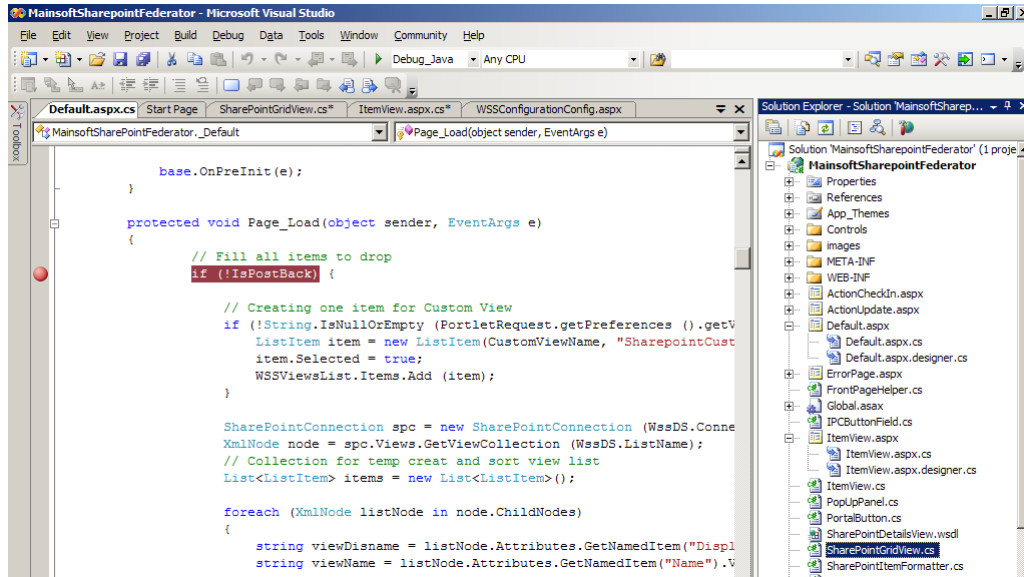


Figure 22. Integrated debugging with the Mainsoft SharePoint Federator

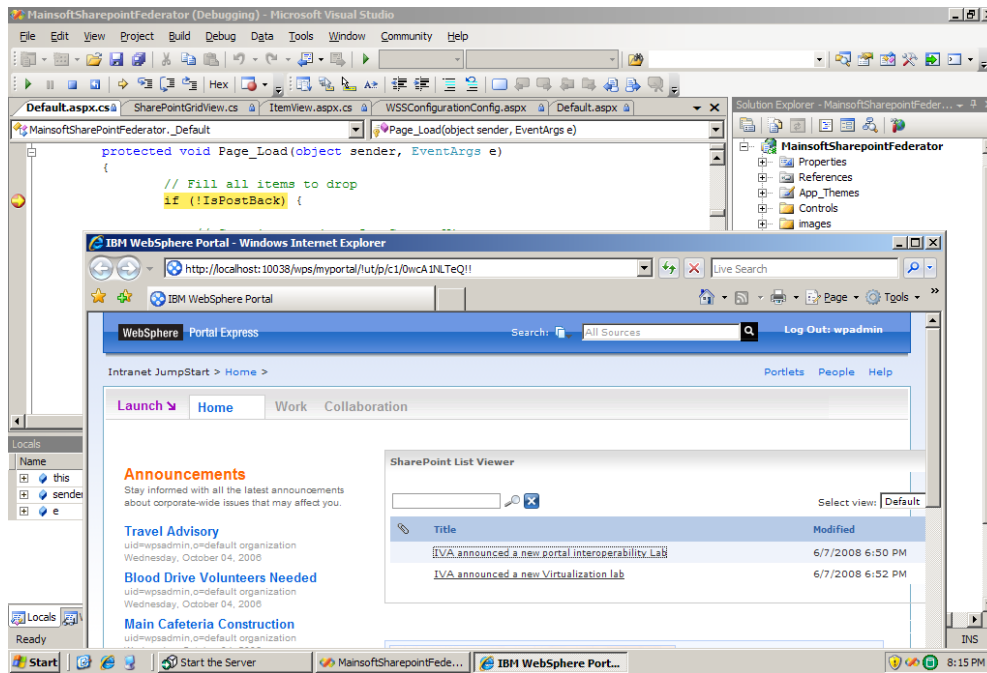


Figure 23. Integrated debugging with the Mainsoft SharePoint Federator

Interoperability between Microsoft Office SharePoint Server 2007 and BEA WebLogic Portal with BEA AquaLogic .NET Application Accelerator

BEA AquaLogic .NET Application Accelerator 1.1 helps organizations reuse existing Microsoft ASP.NET technology and expose it through the Web Services for Remote Portlets (WSRP) standard. Using the BEA solution, developers can integrate portal assets built on the Microsoft .NET Framework into Java-based BEA WebLogic or AquaLogic portals.

In this lab scenario, an organization needs to display information stored on a departmental SharePoint site within its BEA WebLogic Portal enterprise portal. To meet this requirement, we used the BEA AquaLogic.NET Application Accelerator to expose SharePoint lists within portals based on BEA WebLogic Portal 9.2¹. We primarily used the .NET WSRP Producer component included with the application accelerator to expose some ASP.NET pages through the WSRP protocol. This architecture is shown in Figure 24.

In our scenario, the departmental SharePoint site has three lists that the organization wants to expose on the enterprise portal: Announcements, Links, and Calendar. To expose these components, we did the following:

1. Built ASP.NET pages for each list we wanted to expose.²
2. Used open source tools available on [CodePlex](http://www.codeplex.com) (www.codeplex.com) to write the code that extracts data from the SharePoint site to display it on an ASP.NET page.
3. Registered the new pages with the WSRP Producer so they could be exposed to BEA WebLogic Portal as a remote portlet.

¹ The .NET Application Accelerator is not yet supported on WebLogic 10.2.

² Because of the complexity of the SharePoint user interface, it is not possible to expose Web Parts directly through the WSRP Producer that ships with the BEA AquaLogic .NET Application Accelerator because there are several links and resources that would not be handled correctly.

Build a Microsoft ASP.NET page that exposes a SharePoint list

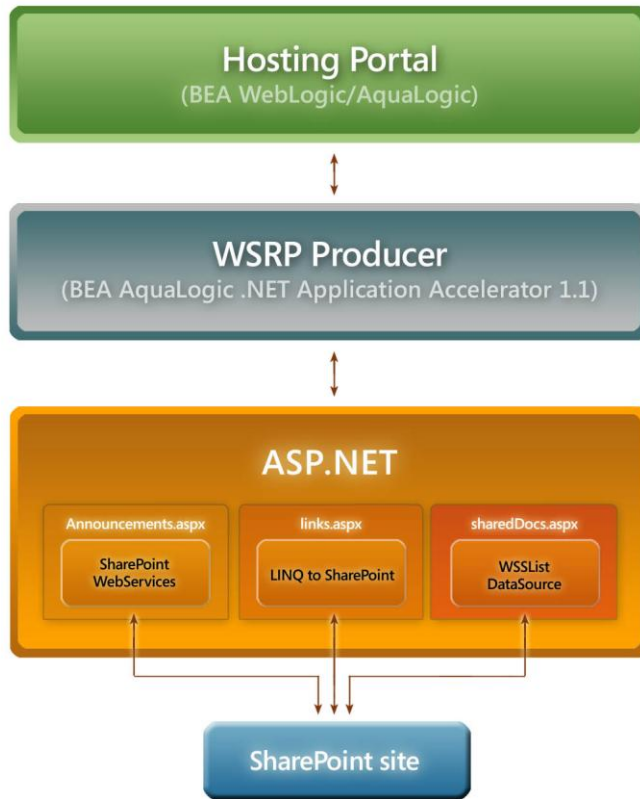


Figure 24. Architecture used in testing the BEA AquaLogic .NET Application Accelerator 1.1 with Office SharePoint Server 2007

We used the Microsoft Visual Studio 2008 development system and ASP.NET to create a Web site with three pages: one page for announcements, another for links, and a third for calendar items.

Extracting and exposing data from SharePoint site pages

To extract and expose data from these SharePoint site pages in the WebLogic portal, we had several options. One option was to use the Web services in Office SharePoint Server 2007 exposed in the _VTI_BIN directory. Office SharePoint Server 2007 exposes a set of Web services to provide access to lists, users, and site configuration settings. Documentation on how to use SharePoint Web services can be found in the Office SharePoint Server 2007 SDK at

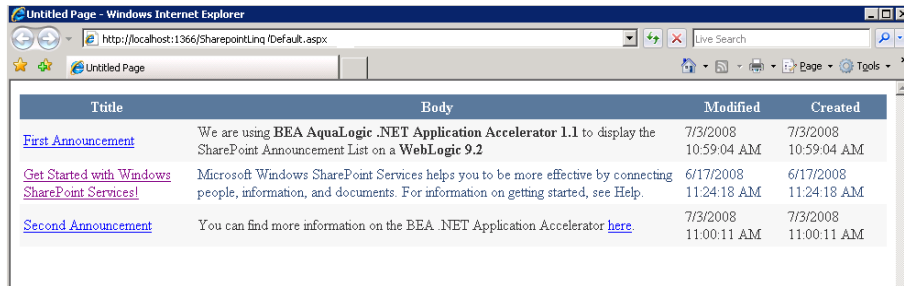
<http://www.microsoft.com/downloads/details.aspx?familyid=6D94E307-67D9-41AC-B2D6-0074D6286FA9&displaylang=en>.

Other options include using components that are built on top of the Web services and can make the coding side quicker and easier. For this lab, we used two components:

- [LINQ to SharePoint project on CodePlex](#). This component allows the developer to build LINQ queries on SharePoint lists.
- [WSS List DataSource on CodePlex](#). This component allows the developer to use a standard DataSource object to connect to a SharePoint list.

Important Note: For both of these CodePlex projects, we recommend against using this code on a production environment. For more critical scenarios, it is recommended to use the SharePoint Web services option instead.

The ASP.NET page is based on a GridView control that shows the relevant fields in the SharePoint list. This is shown below, in Figure 25.



Title	Body	Modified	Created
First Announcement	We are using BEA AquaLogic .NET Application Accelerator 1.1 to display the SharePoint Announcement List on a WebLogic 9.2	7/3/2008 10:59:04 AM	7/3/2008 10:59:04 AM
Get Started with Windows SharePoint Services!	Microsoft Windows SharePoint Services helps you to be more effective by connecting people, information, and documents. For information on getting started, see Help.	6/17/2008 11:24:18 AM	6/17/2008 11:24:18 AM
Second Announcement	You can find more information on the BEA .NET Application Accelerator here .	7/3/2008 11:00:11 AM	7/3/2008 11:00:11 AM

Figure 25. Announcements list from the SharePoint site as displayed on the WebLogic portal

The LinqToSharepoint component makes creating the code to extract the announcements very simple:

```

...
public SharePointList<SharepointLinq.Announcements> announcements;
protected void Page_Load(object sender, EventArgs e)
{
    var context = new SharePointDataContext(new
    Uri("http://iva_bea/SiteDirectory/WSRPDemo"));
    var announcements = context.GetList<Announcements>();
    IEnumerable<Announcements> res = from a in announcements
    orderby a.Title ascending
    select a;
    SharepointLinq.Announcements p = res.ElementAt<Announcements>(0);
    GridView1.DataSource = res;
    GridView1.DataBind();
...

```

The LinqToSharePoint component includes a tool (SPMetal) that is used to export SharePoint list definitions (SharepointLinq.Announcements) to entity classes that can be used for querying using Language Integrated Query (LINQ).

The GridView DataSource property is set to the returned value from the LINQ query that extracts all announcements and sorts them by title in ascending order. More complex queries can be built to make more specific selections, for example, selecting only the announcements with a specific expiration date.

In the DataGrid, the title column was defined as hyperlink so users are redirected to the SharePoint site if they want to edit the item. To make sure the correct URL is used, the DataNavigateUrlFormatString was set to http://iva_bea/SiteDirectory/WSRPDemo/Lists/Announcements/DispForm.aspx?ID={0} (the {0} param is used to specify the ID of the item to display and it is one of the fields retrieved from the SharePoint site). All fields can be easily defined using the columns formatting wizard in the DataGrid, as shown in Figure 26.

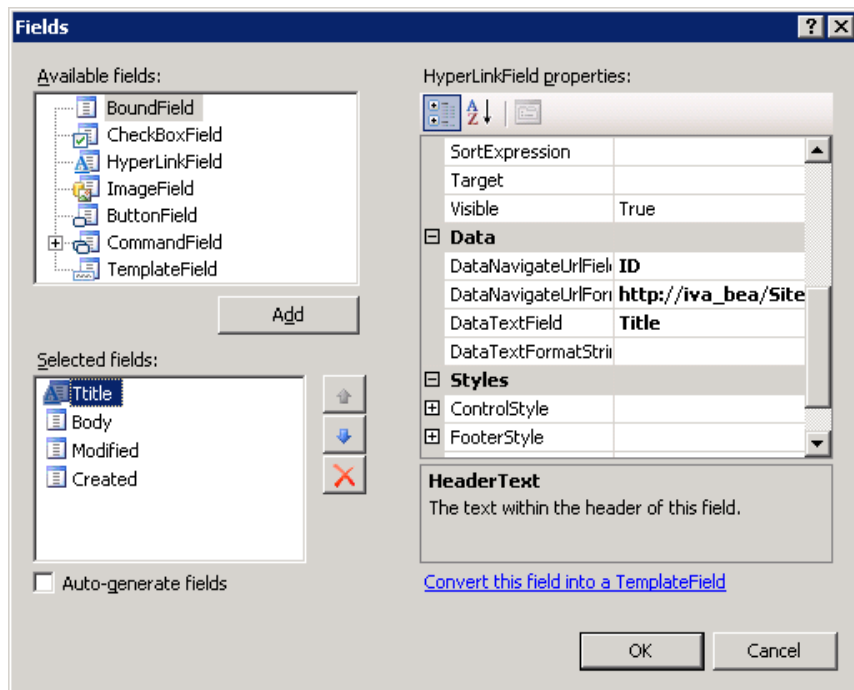
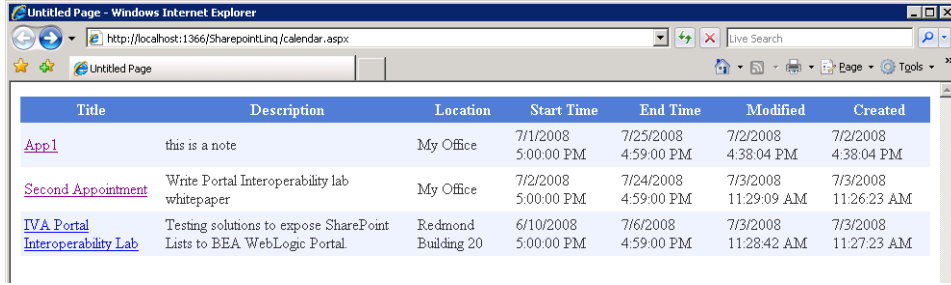


Figure 26. Defining fields in the DataGrid formatting wizard

Through a similar process, we built pages for the links and calendar items, as shown in Figure 27.



Title	Description	Location	Start Time	End Time	Modified	Created
App1	this is a note	My Office	7/1/2008 5:00:00 PM	7/25/2008 4:59:00 PM	7/2/2008 4:38:04 PM	7/2/2008 4:38:04 PM
Second Appointment	Write Portal Interoperability lab whitepaper	My Office	7/2/2008 5:00:00 PM	7/24/2008 4:59:00 PM	7/3/2008 11:29:09 AM	7/3/2008 11:26:23 AM
IVA Portal	Testing solutions to expose SharePoint	Redmond	6/10/2008	7/6/2008	7/3/2008	7/3/2008
Interoperability Lab	Lists to BEA WebLogic Portal	Building 20	5:00:00 PM	4:59:00 PM	11:28:42 AM	11:27:23 AM

Figure 27. SharePoint site page showing calendar list items

An alternative to extracting data from the SharePoint site and building the page is to use the WSS List DataSource component. The main difference with this method is that the WSS List DataSource is a DataSource object that can be added to the ASP.NET page and used directly from a DataGrid.

Figure 28. shows how the WSS List DataSource can be configured to point to the correct list on a SharePoint site.

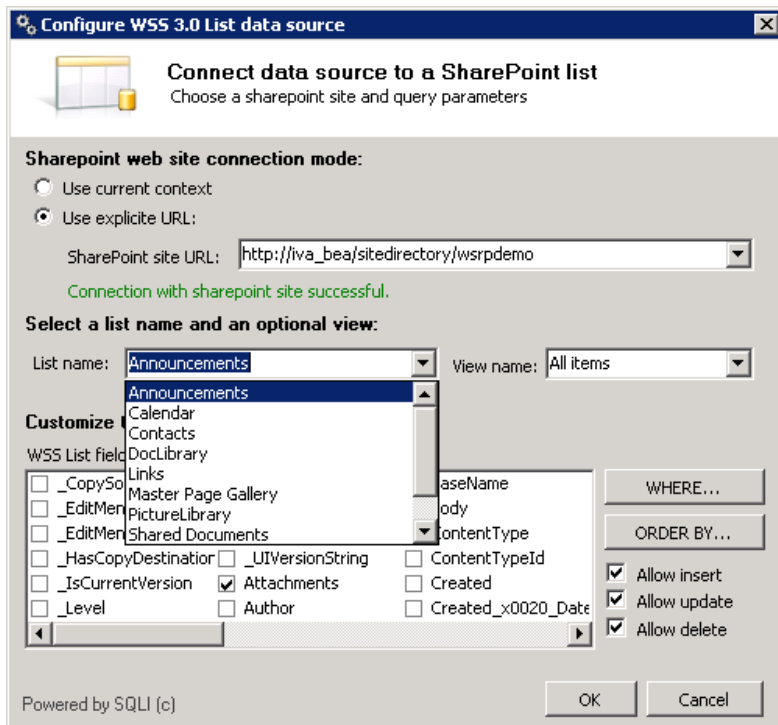


Figure 28. Configuring the WSS List Data source to point to a SharePoint site

Configuring the WSRP Producer

The next step is to configure the BEA AquaLogic .NET Application Accelerator WSRP Producer so that it lists the new site pages as remote portlets.

WSRP Producer configuration

There are few things that need to be configured on the WSRP Producer side to make sure that the correct credentials are used to connect to the SharePoint site. The ASP.NET pages rely on the user credentials to access the SharePoint site. When the pages are accessed by the WSRP Producer, the user might change—that is, the user that is connecting to the WSRP Producer could be different from the user who is authorized to access the SharePoint site—so we needed to ensure that the correct credentials were passed from the WebLogic Server to the WSRP Producer and then used to access the ASP.NET page so that it could connect to the SharePoint site to retrieve the items in the lists.

Enabling UNT Authentication

Enable UNT authentication on the WSRP Producer, which requires updating the WSRPService.WSDL and WSRP_V1_Bindings.wsdl files (SAML can also be used).

Defining Credential Mapping

Define a Credential Mapping on BEA WebLogic Portal side so that the user name used to access to the portal is mapped to a user that can access the SharePoint site.

Detailed instructions on how to configure BEA WebLogic Portal are available at http://e-docs.bea.com/en/alui/dotnetappaccelerator/docs11/WSRPDevelopmentGuide_ALDotNetApplicationAccelerator/tsk_wsrp_authoring_managingwlpauth_unt_iis.html.

Hardcoding Credentials, an alternative to WSRP Producer Configuration

Credentials can be hardcoded in the code that is used to access the SharePoint site. Even though it is not recommended, because it presents a security risk, the following code shows how credentials can be used from the LinqToSharepoint component:

```
...
protected void Page_Load(object sender, EventArgs e)
{
    var context = new SharePointDataContext(new
    Uri("http://iva_bea/SiteDirectory/WSRPDemo"));
    context.Credentials = new NetworkCredential("administrator", "password",
    "iva_bea");
}
```

...

Adding the new pages to the list of ASP.NET pages that can be exposed

After the security was configured, we added the new pages to the list of ASP.NET pages that the WSRP Producer can expose as a remote portlet. This can be done by updating the WSRP-Producer.xml file in the WSRP Producer install directory (usually %BEA_HOME%\alui\wsrpproducer\1.1\webapp\wsrpproducer).

A new portlet element was added for each page that needs to be exposed. Here is an example:

```
...
<portlet>
<handle>WSRPLinqCalendar</handle>
<!-- urls specified here are available for any mode to reference -->
<url id="view">http://IVA_BEA:112/calendar.aspx</url>
<url id="edit">http://IVA_BEA:112/calendar.aspx</url>
<expiration-cache>10</expiration-cache>
<supported-locale>en</supported-locale>
<description lang="en">Calendar Item list</description>
<display-name lang="en">WSRPLinqCalendar</display-name>
<supports>
<mime-type>text/html</mime-type>
<portlet-mode>
<name>wsrp:view</name>
<!-- url is omitted so the first url for this portlet is used -->
</portlet-mode>
<portlet-mode>
<name>wsrp:edit</name>
<url idref="edit"/>
</portlet-mode>
<portlet-mode>
<name>wsrp:help</name>
<!-- urls may also be specified in-line-->
<url>http://IVA_BEA:112/calendar.aspx</url>
</portlet-mode>
</supports>
<portlet-info>
<title lang="en">WSRPLinqCalendar</title>
<short-title lang="en">View</short-title>
</portlet-info>
<portlet-properties>
<property name="LastAccessed" type="xs:dateTime"/>
<property name="PreferredFormOfAddress" type="xs:string" defaultValue="Mr."/>
</portlet-properties>

```

```
<user-profile>
<item>UserProfileName/PropertyName</item>
</user-profile>

<!-- If this portlet uses forms authentication you can configure automatic
sign-on below -->
<!--
<forms-authentication>
<IsSSOConfigured>>true</IsSSOConfigured>
<loginpage>login.aspx</loginpage>
<username-field-name>userNameTextBox</username-field-name>
<password-field-name>passwordTextBox</password-field-name>
<login-button-name>submitButton</login-button-name>
<login-button-value>Login</login-button-value>
<always-use-anonymous-login>>false</always-use-anonymous-login>
<anonymous-username>uerid</anonymous-username>
<anonymous-password>pwd</anonymous-password>
</forms-authentication>
-->
</portlet>
...
```

Detailed instructions on how to update the WSRP-Producer.xml file can be found at http://e-docs.bea.com/en/alui/dotnetappaccelerator/docs11/WSRPDevelopmentGuide_ALDotNetApplicationAccelerator/ref_wsrpproducer_configurationfile.html.

Creating new remote portlets on BEA WebLogic Portal

After we configured the WSRP Producer, it was time to add the new pages to the WebLogic portal by creating new remote portlets that point to the WSRP Producer previously configured.

We created a new remote portlet by selecting the WSRPService.WSDL file for the WSRP Producer. In this case, it exposed two portlets (ASP.NET pages).

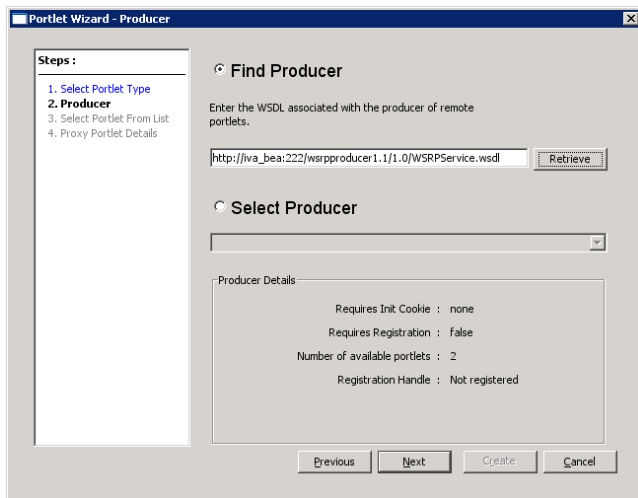


Figure 29. Creating a new remote portlet

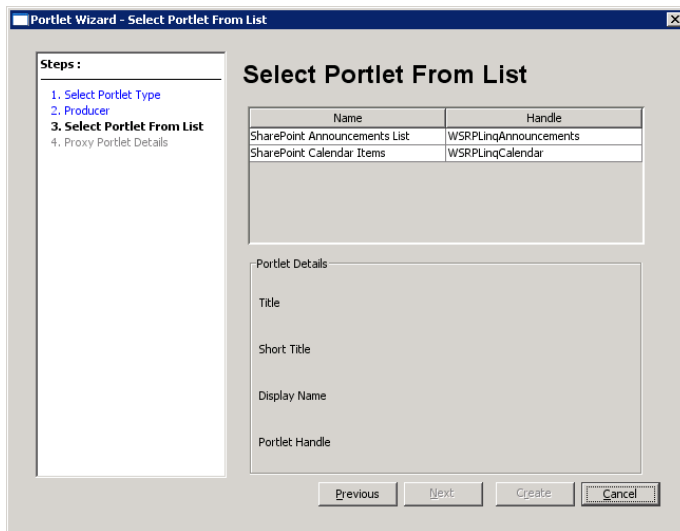


Figure 30. Selecting which portlet to expose from the list of created portlets

Lists keep a reference URL to the SharePoint site so that users can edit the items by clicking on the hyperlinks in the list, as shown in Figure 31.



Figure 31. SharePoint list exposed on WebLogic portal with hyperlinks to items on the site

Appendix: The Interop Vendor Alliance

The Interop Vendor Alliance (IVA) is a cross-industry group of global software and hardware vendors that share the common goal of making mixed technologies work better together for their customers. When people work together, the solutions they design are more likely to work together as well. The IVA provides a venue for its members to share customer feedback and to tackle common interoperability challenges. Through member meetings, hands-on labs, and online forums, IVA members work together to solve common interoperability challenges while improving the effectiveness of their own solutions. Scenarios addressed by the IVA include system management, virtualization, identity management, data integration, storage management, interoperability of developer tools, in addition to portal integration.

The IVA supports:

- *Vendor collaboration to foster interoperability*
- *Scenario-based testing for interoperability*
- *Communicating with customers about interoperability solutions*

Related Links

To learn more about the Interop Vendor Alliance, visit

<http://interopvendoralliance.org/>

To learn more about Microsoft Office SharePoint Server 2007, visit

<http://www.microsoft.com/sharepoint/default.msp>

To learn more about Mainssoft, visit

<http://www.mainssoft.com/>

To learn more about the Mainssoft SharePoint Federator, visit

http://www.mainssoft.com/solutions/pdfs/FederatingSharepointSites_WP.pdf

To learn more about BEA, visit

<http://www.bea.com/>

To learn more about the BEA AquaLogic .NET Application Accelerator, visit

<http://www.bea.com/framework.jsp?CNT=index.htm&FP=/content/products/more/accelerator/>

Table of Figures

Figure 1. Interoperability between Microsoft Office SharePoint Server 2007, BEA WebLogic Portal, and IBM WebSphere Portal.....	6
Figure 2. List Viewer portlet and Item Viewer portlet appearing on WebSphere portal, using SharePoint-based Web services and the WebSphere Credential Vault	8
Figure 3. Installing runtime components.....	8
Figure 4. WebSphere Credential Vault	9
Figure 5. WebSphere portal home page	10
Figure 6. Edit Layout page in WebSphere portal	10
Figure 7. Selecting SharePoint List Viewer.....	11
Figure 8. SharePoint List Viewer portlet added.....	11
Figure 9. Entering URL for document library and user credentials for authentication.....	12
Figure 10. Designating which SharePoint list columns to display in the WebSphere portal	13
Figure 11. Choosing CSS formatting and styles for all List Viewer modes.....	13
Figure 12. Document Library as it appears on the SharePoint site.....	14
Figure 13. WebSphere portal home page showing the document library from the SharePoint site, with document names and details.....	14
Figure 14. Adding the Item Viewer portlet.....	15
Figure 15. Linking the Item Viewer portlet to the List Viewer portlet.....	15
Figure 16. Designating source portlet, target portlet, and data to be exchanged	16
Figure 17. List Viewer portlet with corresponding properties in the Item Viewer	17
Figure 18. Classes exposed by SharePoint Federator	18
Figure 19. Classes available in the Mainsoft SDK to control access to a SharePoint site	19
Figure 20. List of classes included in the SDK to control aspects such as Security, User Profile, and hosting environment.....	20
Figure 21. WYSIWYG editor for managing portlets	20
Figure 22. Integrated debugging with the Mainsoft SharePoint Federator	21

Figure 23. Integrated debugging with the Mainsoft SharePoint Federator	22
Figure 24. Architecture used in testing the BEA AquaLogic .NET Application Accelerator 1.1 with Office SharePoint Server 2007	23
Figure 25. Announcements list from the SharePoint site as displayed on the WebLogic portal .	25
Figure 26. Defining fields in the DataGrid formatting wizard.....	26
Figure 27. SharePoint site page showing calendar list items	27
Figure 28. Configuring the WSS List Data source to point to a SharePoint site	28
Figure 29. Creating a new remote portlet	33
Figure 30. Selecting which portlet to expose from the list of created portlets.....	33
Figure 31. SharePoint list exposed on WebLogic portal with hyperlinks to items on the site.....	35

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This White Paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

This documentation is licensed to you under the Creative Commons Attribution-Share Alike 3.0 United States License.



To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/3.0/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA. When using this documentation, provide the following attribution: Portal Aggregation and Integration Lab - Interoperability between Microsoft Office SharePoint Server 2007, BEA WebLogic Portal and IBM WebSphere Portal (with Mainsoft).

Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, email address, logo, person, place or event is intended or should be inferred.

© 2008 Microsoft Corporation.

Microsoft, SharePoint, and Visual Studio are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.