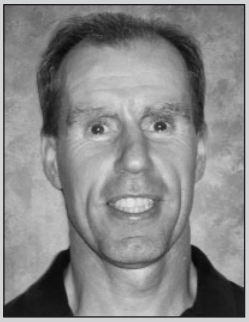
Integrating Non-SAP Data and Web Content into Your SAP Enterprise Portal — A Guided Tour of Your Options, Which to Use, and When

Patrick Dixon



Patrick Dixon is a manager with Deloitte specializing in SAP Enterprise Portal and Master Data Management (MDM). He has over five years of experience Web-enabling and integrating SAP systems with Enterprise Portal, ITS, Exchange Infrastructure (XI), and Plumtree Software solutions. Patrick was a key speaker at SAP BW and Portals 2004 and SAP Portals 2003.

(complete bio appears on page 118)

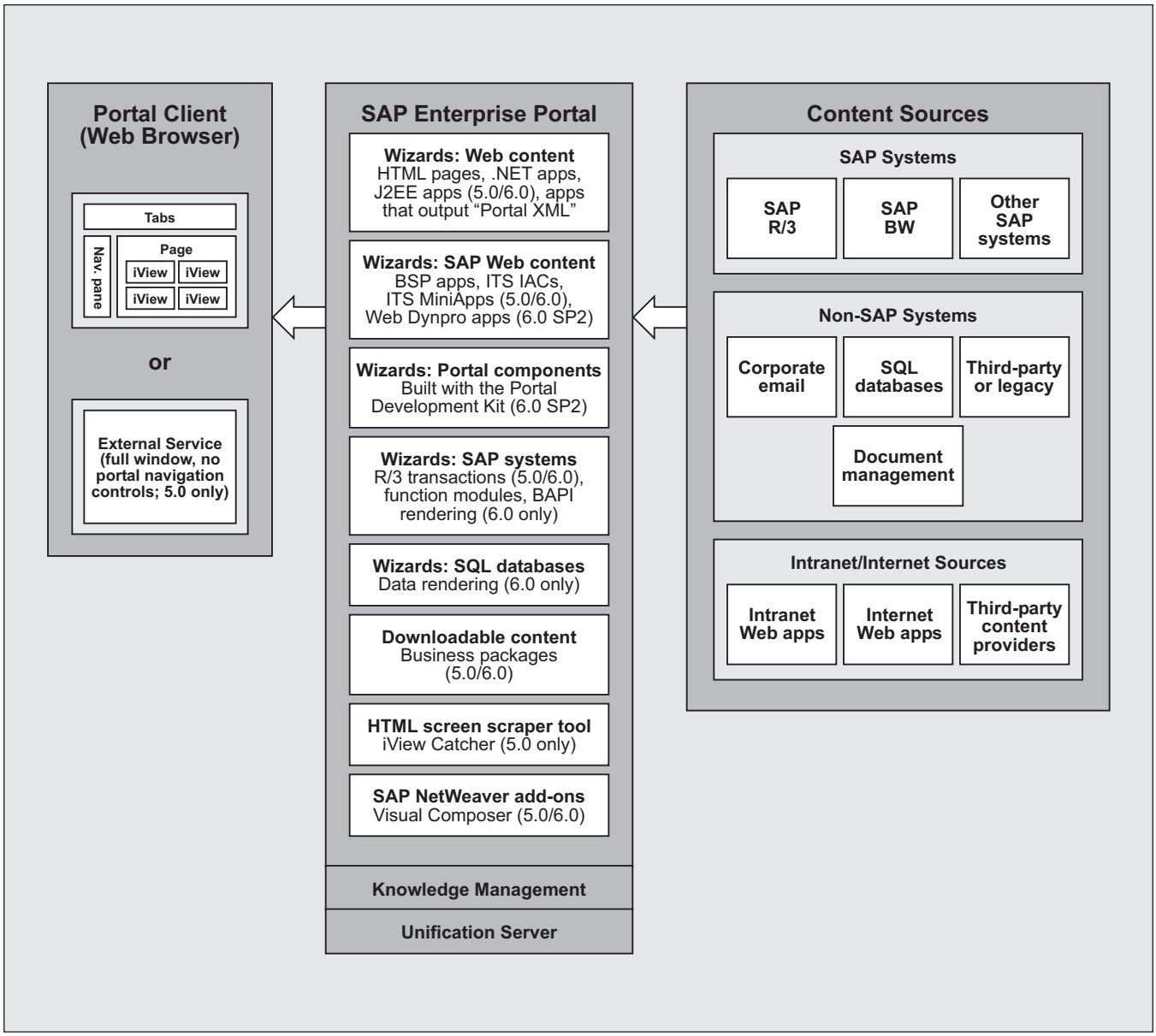
Any user — whether an end user, administrator, or developer — can relate to the tediousness of navigating among the various applications and information locations, both inside and outside of the organization, needed for daily tasks. A portal that offers a single point of seamless access to various internal and external systems and content can save users a significant amount of time and frustration, and improve productivity. To achieve these benefits, however, users need to actually use the portal, and the key to getting users to use the portal is providing them with the content that they need — in fact, I have found that the right information in the right place is the single most important factor in user acceptance. SAP Enterprise Portal (SAP EP) provides you with the tools you need to provide a powerful, personalized portal for your users that will become a natural, centralized workplace for them.

The first installment in this article series showed you how to use the wizards, tools, and predeveloped business packages provided for SAP EP 5.0 and 6.0 to integrate transactions, reports, and data from your SAP systems, including SAP R/3 and SAP BW.[1] This second installment shows you how to bring your non-SAP systems into the fold.

First I'll review the main details you need to know about the SAP EP toolset, including the differences between the SAP EP 5.0 and 6.0 architectures and iView administration models, as well as SAP EP's key content integration capabilities. Then, I will walk you through the integration options for non-SAP systems, including corporate email systems and third-party SQL databases. Finally, I will show you how to bring Web-based content from intranet and Internet sources into your portal.

[1] "Integrating SAP Transactions, Reports, and Data into Your SAP Enterprise Portal — A Guided Tour of Your Options, Which to Use, and When" (*SAP Professional Journal*, July/August 2004).

Figure 1 The SAP EP Toolset for Integrating SAP, Non-SAP, and Web Content



SAP Enterprise Portal Fundamentals

The first installment of this article series provides a lot of important foundational knowledge that will help you get the most out of this second installment. While I recommend reading the previous article first, for your convenience I've summarized the key points of that article in the following sections.

The SAP Enterprise Portal Toolset

Figure 1 summarizes the SAP EP toolset and its integration capabilities. As you can see, the tools include:

- Generic wizards for integrating Web-based content and applications
- Specialized wizards for integrating Web-based applications built with SAP technologies

> ✓ ***Note!***
>
> *The optional Knowledge Management and Unification components shown in Figure 1 are beyond the scope of this article series, which focuses on integrating content with the base SAP EP server configuration. Knowledge Management provides a content management system for SAP EP. Unification's "Drag&Relate" functionality enables you to define hotspots on portal pages onto which users can drag and drop objects (e.g., a user can drag an order number onto a "Display Order" link that launches the Display Order transaction). It can also automatically render HTML views of SAP R/3, SAP BW, and Microsoft SQL Server data. Through SAP EP 6.0 SP2, the Unification server only runs on Microsoft Windows, and requires access to a Microsoft SQL Server system. In SAP EP 6.0 SP3,[2] the Unification server has been integrated into the core SAP Web Application Server-based portal server platform. For more on these add-ons, visit the SAP online help at http://help.sap.com/portals.*

- Wizards for integrating portal components built with the Portal Development Kit (PDK)[3]
- Specialized wizards for integrating SAP systems and SQL databases
- Predefined business packages (available at the Portal Content Portfolio[4] Web site) containing iViews, pages, and roles developed by SAP and SAP partners that can be downloaded and integrated
- An HTML "screen scraper" tool (available with SAP EP 5.0 only) for extracting and integrating content from Web pages and Web-based applications
- SAP NetWeaver add-on tools, such as the new Visual Composer, which renders a customizable HTML interface to backend data

The first installment in this article series provided a detailed overview of each of these powerful SAP EP 5.0 and 6.0 integration tools.

The SAP Enterprise Portal Architecture

SAP EP 5.0 and 6.0 have radically different architectures. While both SAP EP 5.0 and 6.0 run on a Java platform, SAP EP 5.0 requires two servers — a Microsoft Internet Information Server (IIS) as its Web server, and an SAP J2EE server for the portal runtime — and runs only on Windows systems. With SAP EP 6.0, the portal runs on SAP Web Application Server (SAP Web AS) with the Java runtime installed, which includes a new proprietary Web server as well as a J2EE server (the SAP J2EE Engine).

This architectural change has had three main impacts:

1. You can host SAP EP 6.0 on either Unix or Windows systems.
2. SAP EP 6.0 cannot run Active Server Page (ASP) and .NET applications on its own, since it uses a proprietary Web server instead of a Microsoft IIS Web server. ASP/.NET applications can only be hosted on your SAP EP 6.0 portal server if it is running on Windows and has Microsoft IIS running in parallel to the portal Web server.[5]
3. Tools and business packages that rely on the Microsoft-based SAP EP 5.0 architecture —

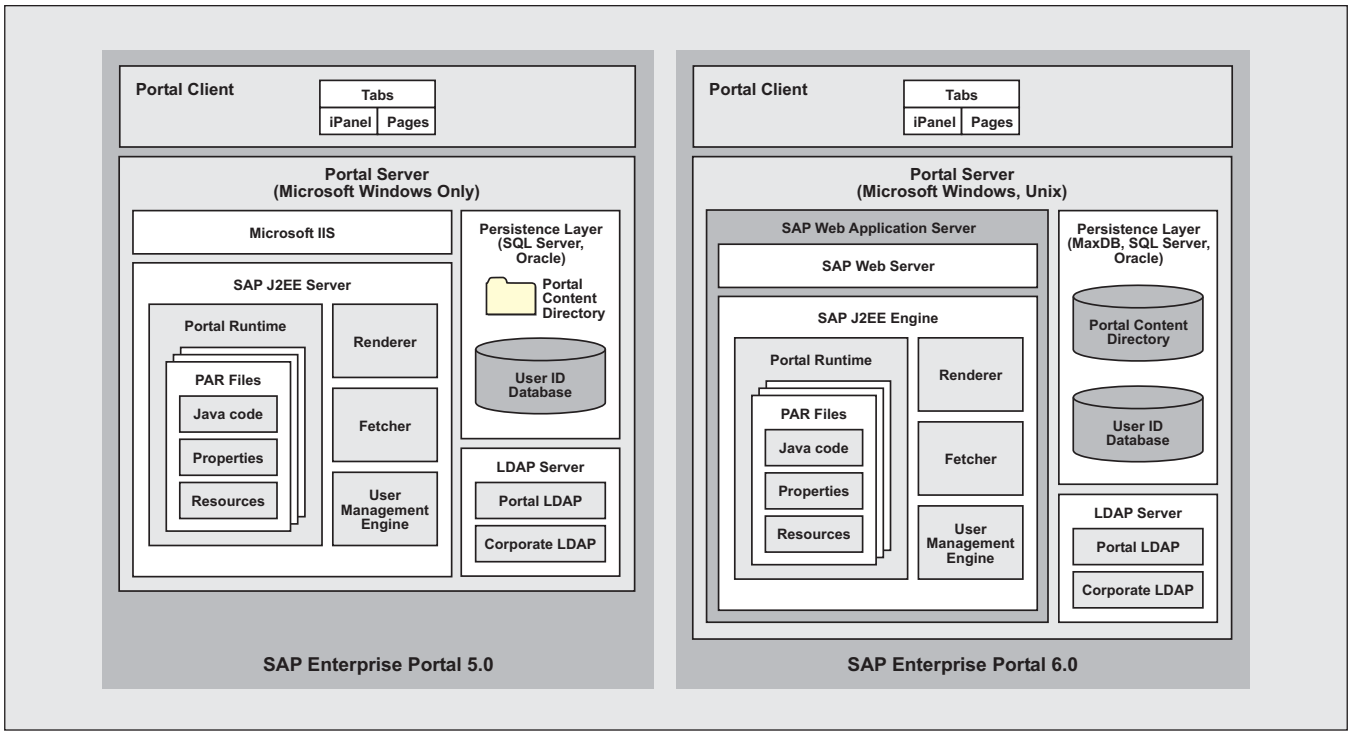
[2] SAP EP 6.0 SP3 has been renamed "SAP EP 6.0 running on Web Application Server 6.40" to highlight their relationship. In this article, I continue to call it SAP EP 6.0 SP3 for simplicity and continuity with my previous article.

[3] The PDK is available for download from http://sdn.sap.com.

[4] Note that the iViewStudio has been renamed the "Portal Content Portfolio" and has moved to SDN (https://www.sdn.sap.com/sdn/contentportfolio.sdn).

[5] For details on setting up Microsoft IIS as a Web proxy to run ASP/.NET applications, see the article "Designing a Solid, Lasting Landscape for Your SAP Enterprise Portal Implementation: Using the Most Effective Technical Options to Meet Your Key Requirements" (*SAP Professional Journal*, November/December 2004).

Figure 2 The SAP EP 5.0 and 6.0 Architectures



like the HTML screen scraper tool, which uses Microsoft components, and the Communication business package, which contains iViews for accessing mail from Microsoft Exchange — are no longer supported. Some business packages, like the CRM business package, have been ported to SAP EP 6.0 and rereleased in the Portal Content Portfolio. Other 5.0 business packages, though not "officially" supported, will continue to run on 6.0, though only trial and error can determine which.[6]

Figure 2 provides a graphical overview of the two architectures.

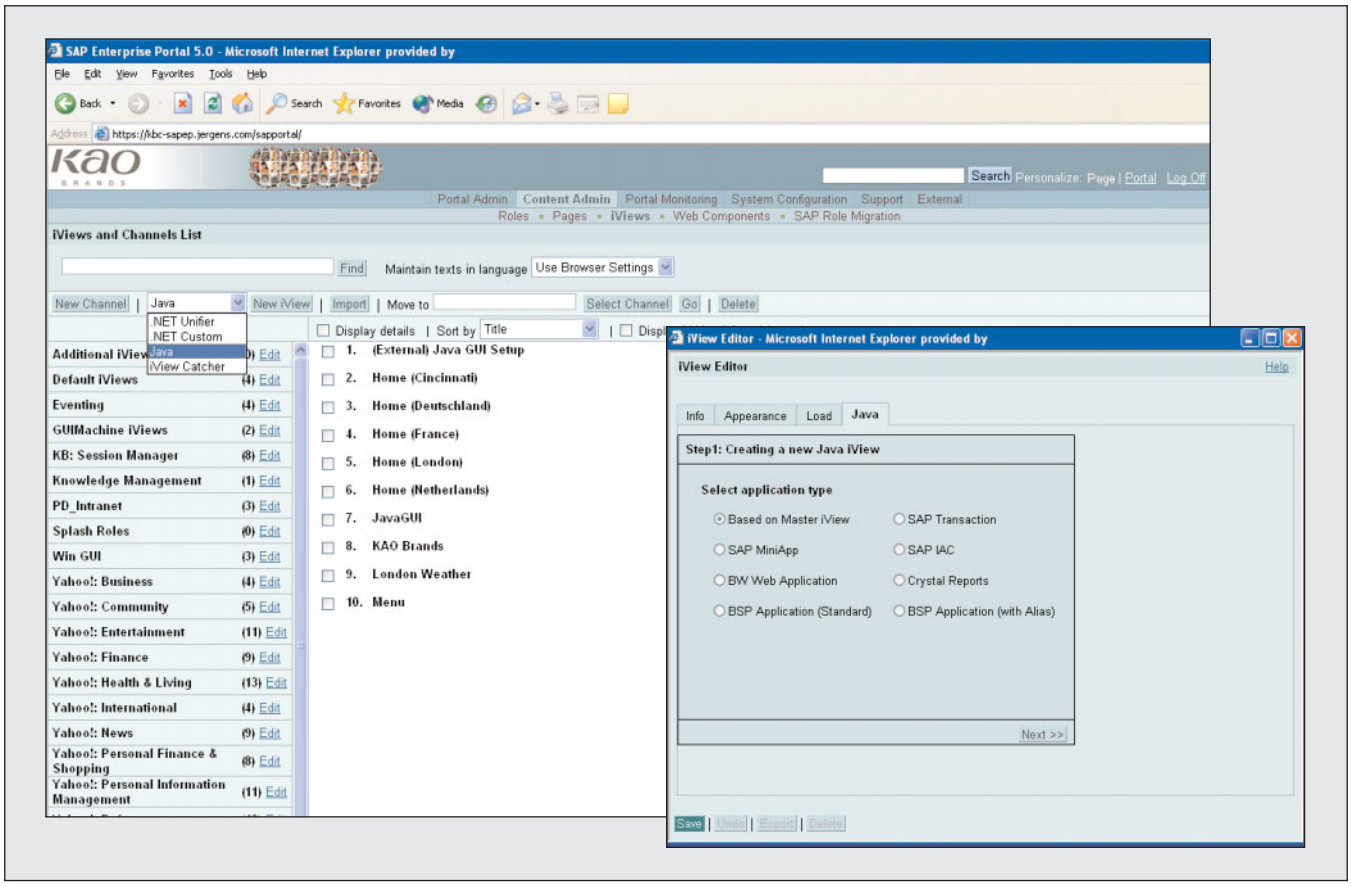
As you can see in Figure 2, another key difference between the two architectures is the storage of persistent data — data that remains when a user logs off and back on to the portal. In SAP EP 5.0, the data is stored in different locations: for example, user information like ID, email addresses, contact information, and account validity are stored in a separate portal database, while information such as iView content, role definitions, page-to-role relationships, and personalization data are referenced from the Portal Content Directory (PCD) in the portal server file system. Storing data in more than one place is confusing, however, and updates to the portal server can adversely affect the files in the PCD. SAP EP 5.0 also requires a locking server to coordinate concurrent file access if portal servers are clustered. SAP EP 6.0 resolves these problems by storing all the persistent data, including the PCD, centrally in the portal database, and does not require a separate locking server.

SAP Enterprise Portal Administration

In addition to (and partly as a result of) the architecture change between SAP EP 5.0 and 6.0, SAP EP 6.0 reorganizes the navigation structure of the Web-based portal administration tool for creating, managing, and editing iViews, pages, and roles.

[6] Note that once installed, business packages cannot be uninstalled, so it's best to try them out on a sandbox system first, to avoid cluttering your development system.

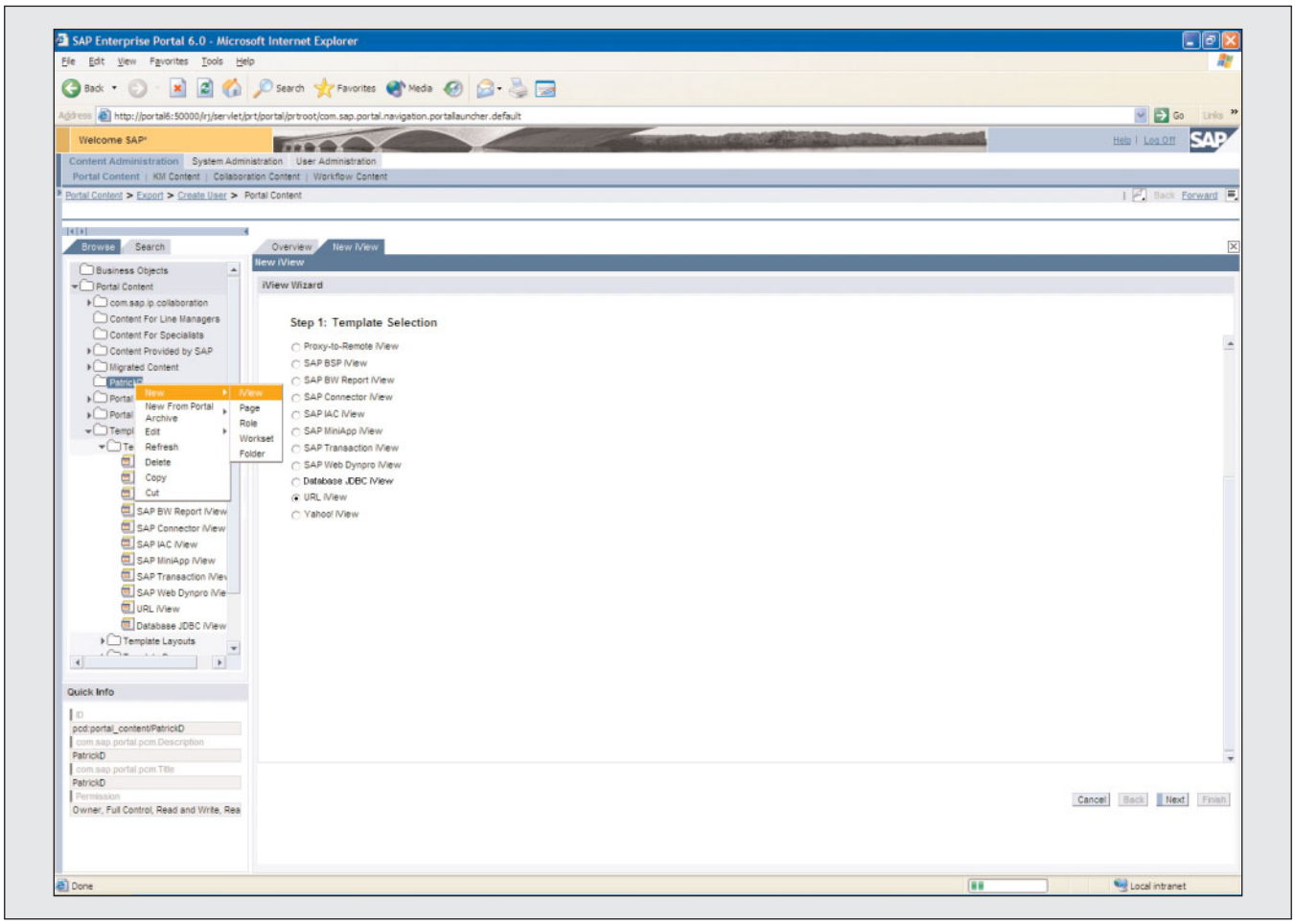
Figure 3 Creating an iView in SAP Enterprise Portal 5.0



> ✓ ***Note!***
>
> *SAP also renamed a few key iViews in SAP EP 6.0. For example, the oddly named ".NET Custom iView" in 5.0 has been renamed "URL iView" in 6.0 — its function is the same, but the name reflects its purpose more clearly. In the diagrams in this article, I've placed functionally analogous iView types next to each other and flagged them with the version they apply to.*

In SAP EP 5.0, you first choose the type of iView you want to create from the dropdown shown at the upper left in **Figure 3**. Selecting the iView Catcher item from the dropdown launches the screen scraper tool for capturing Web page content. Selecting the .NET Unifier, .NET Custom, and Java items launches a dialog-based iView Editor wizard for defining that particular iView type. The .NET Unifier option is for creating Drag&Relate iViews (see the note on page 87). The most frequently used 5.0 iView type is .NET Custom, which is for creating iViews based on URLs, but most of the useful integration options — for integrating SAP transactions and BW Web reports, for example — are located under the Java iViews option; Figure 3 shows the Java iView subtypes that enable you to integrate various types of content.

Figure 4 Creating an iView in SAP Enterprise Portal 6.0



SAP EP 6.0 simplifies iView administration by replacing the different iView types and subtypes with "templates" (see **Figure 4**). To create a new iView, you simply navigate to a folder in the PCD and double-click on an iView template (e.g., SAP Transaction iView), or right-click on the target folder for the new iView and select *New → iView*.

Key SAP Enterprise Portal Content Integration Capabilities

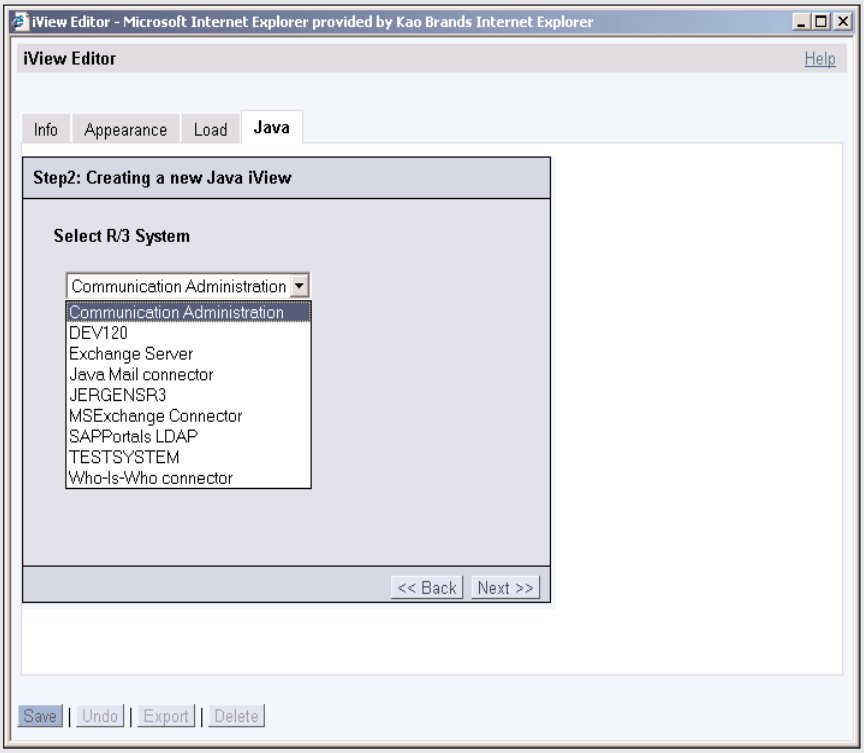
The first installment in this article series walked you through the available options for integrating content into your portal, with a particular focus on integrating content from your SAP systems (see the sidebar on the next page for a review of the system settings that must be in place to enable portal content integration). The key integration capabilities to remember are:

- **The Portal Content Portfolio offers several important business packages for SAP content integration.** Business packages provide prefabricated portal components, such as iViews that link to backend systems, user roles, and portal pages. SAP has worked with many different partners to produce business packages that render their applications and content into a portal. Business packages are quick to install and inexpensive to deploy. Particularly useful packages for SAP system integration, which I discussed in the previous article, include the Employee Self-Service (ESS)

Prerequisite Setup for Content Integration

As shown in the screenshot below, many of the portal's wizards will ask you to select from a list of backend systems rather than let you specify connection criteria explicitly.

This list is drawn from a repository called the Portal System Landscape, which must be maintained by your portal administrator. The procedure depends on which SAP EP version you are running.



In SAP EP 5.0, system definitions are stored in two files — systems.xml and JCODestinations.xml — which you'll find in the file-system-based Portal Content Directory. To add or maintain the available systems, you'll need to download, edit, and upload these files. For convenience, the portal administration tool (*Configuration → Systems Landscape*) includes pushbuttons for downloading and uploading the files.

In SAP EP 6.0, system definitions are all stored in a file called portalapp.xml, which is stored in compressed format in the Portal Content Directory database. It cannot be accessed or edited directly. Instead, SAP has added an administration tool (*System Administration → System Configuration → System*) to visually add new systems — no manual edits to the XML files required!

> ✓ ***Tip***
>
> *Be forewarned! Editing the systems.xml and JCODestinations.xml files requires a basic understanding of XML and a steady hand. Examples of each file are available for download at www.SAPpro.com — the rest is up to you. Be sure to make a backup copy of each file before making any modifications, and keep the backups on hand until well after you've tested your changes.*

business package, which contains iViews that let employees update their contact information in SAP R/3 HR, and the Customer Relationship Management (CRM) business package, which contains over 1,000 roles, pages, and iViews for customers and employees to access SAP CRM. I'll introduce some important business packages for non-SAP and Web content later in this article.

- **Any Web-based resource can be instantly brought into your portal.** The rule to remember is that if it's accessible via a URL — like static Web pages, .NET or J2EE applications, and Web-enabled SAP R/3 reports — you can bring it into a portal iView either explicitly with the URL iView template in SAP EP 6.0 (.NET Custom iView in SAP EP 5.0.), or using one of the "higher-level" iView types (e.g., the BW Report iView template in 6.0 or the Java iView subtype BW Web Application in 5.0).[7] Regardless of the iView template you use, the portal ultimately connects to all Web-based resources via HTTP and requests the resource via a URL. The only difference with the higher-level wizards is that the iView knows which URL parameters the target system is expecting. In my previous article, I discussed your options for integrating existing Web-based SAP applications. In this article I'll discuss your options for integrating non-SAP Web-based applications.

> ✓ ***Note!***
>
> *I will use the terms "wizard" and "template" interchangeably in this article. While it is more accurate to say that an SAP EP 6.0 iView is created based on a template, the ensuing process is still wizard-based.*

- **You can deploy nearly any SAP transaction or report in your portal without modification.** Using the SAP Transaction iView, simply enter the SAP transaction code for the transaction or report, specify which SAPGUI the iView should use — SAPGUI for Windows (WinGUI), SAPGUI for Java (JavaGUI), or SAPGUI for HTML (WebGUI) — and the portal launches the GUI within the user's browser. To use WinGUI, it must already be installed on the portal user's desktop.[8] In contrast, the JavaGUI option can be either installed on the PC in advance or downloaded at runtime — the iView automatically detects whether the JavaGUI is installed on the client and installs it if needed.[9] The WebGUI option is appropriate if you have (or are willing to install) SAP Internet Transaction Server (ITS). WebGUI dynamically generates Web pages that look and function just like WinGUI or JavaGUI without the need to install anything on user PCs.[10]

- **The Visual Composer tool and the SAP Connector iView template let you directly render SAP data.** If you can find a BAPI (or any other function module) that outputs the data you want (or that can create the data you want), these two tools will automatically generate a Web frontend for you. The Visual Composer is a WYSIWYG tool, and is distributed as an add-on to your SAP EP 5.0 or 6.0 portal.[11] Upcoming Visual Composer releases will include connectors

[7] This doesn't mean the content or application moves to the portal. It just means the portal requests the content from the backend system at runtime and directs the HTML result to the iView container on the portal page.

[8] You can also configure the portal to launch WinGUI natively outside the browser to make SAPGUI "power users" more comfortable. See my previous article in the July/August 2004 issue for more on how to do this.

[9] The first JavaGUI installation can take some time as it is a 15 MB file, but it only needs to be installed once.

[10] WebGUI is a bit slower than WinGUI and JavaGUI, and there are a few minor technical differences. For example, value lists (F4 pop-up dialogs) don't come through as pop-ups — instead, they replace the current Web page in the main browser and return the user to the main page when he or she selects a value.

[11] The Visual Composer tool was formerly known as the "GUI Machine," which is why you will often see references to "GUI Machine" or "GM" when using or administering the tool. As of this writing, the 6.0 version of Visual Composer is still in pilot release. Check http://sdn.sap.com for the most recent availability.

> ✓ ***Note!***
>
> *You may have heard that ITS has been integrated into SAP Web AS 6.40, which underlies SAP EP 6.0 SP3. While this is true, do not decommission your standalone ITS servers just yet!*
>
> *First, the ITS built into SAP Web AS 6.40 (called "Integrated ITS") can only access the ABAP runtime of the system on which it is installed. This means that you won't be able to use the ITS within your portal SAP Web AS to run transactions on your SAP R/3, SAP BW, or SAP CRM systems, for example. I imagine SAP will address this requirement in the next version since this is an important requirement.*
>
> *Second, only ITS's WebGUI feature and Easy Web Transaction (EWT) — also known as Internet Application Component (IAC) — development model have been ported to SAP Web AS. If you use ITS's Web reporting feature or have applications that rely on ITS's WebRFC or Flow Logic development models, you will still need your standalone ITS.*[12]

to non-SAP systems as well (e.g., via JDBC), which I'll discuss later in this article. The SAP Connector iView is built into SAP EP 6.0 only, but doesn't have a visual design tool — it generates an automatic, noncustomizable interface based on the list of input/output parameters you choose to include.

With this review fresh in mind, let's now discuss how to bring existing, non-SAP data into your portal. We'll then tackle Web-based content from intranet and Internet sources.

[12] This was intentional: EWTs and Flow Logic have been replaced by Business Server Pages (BSP) and Web Dynpro in SAP Web AS, and ITS Web reporting had some technical limitations that motivated SAP to recommend that even ITS run reports with WebGUI.

Integrating Existing, Non-SAP Content with Your Portal

In many ways, integrating non-SAP content is similar to integrating SAP content. There are four key strategies:

- **Use a business package.** As with SAP systems, the best and cheapest way to integrate non-SAP content is to leverage the business packages available in the Portal Content Portfolio. You'll find packages from SAP and third-party vendors, and occasionally ones that customers have chosen to share. Initially, SAP authored many packages and connectors to third-party systems — e.g., Microsoft Outlook and FedEx — to "get the ball rolling." Circa SAP EP 6.0, however, a large number of third-party vendors have released their own specialized business packages that in most cases supplant the older SAP ones. In light of this shift, SAP has chosen to discontinue (or "sunset") their non-SAP business packages, which is why you'll see the Portal Content Portfolio awash in non-SAP business packages for SAP EP 5.0 only. In some cases, SAP has even pulled useful "generic" connectors from the 5.0 business packages (e.g., the JavaMail service from the email business package, which we'll explore later) and integrated them into the core 6.0 portal platform.

> ✓ ***Tip***
>
> *When searching for business packages in the Portal Content Portfolio, be aware that only SAP-provided packages appear on the default screen. Be sure to also click on the Vendor Content and Customer Share tabs to see packages provided by vendors or other customers. For example, at the Portal Content Portfolio's SDN site, go to Portal Content Catalog → Search Catalog, then click on the Content and Documentation link in the left navigation pane.*

- **Web-enable and integrate.** As Web technologies have become more prevalent, many vendors have released Web-enabled (or Web-native) versions of their systems. For example, a few years ago IXOS embedded a Web server into its archiving product, and ported all of its frontend applications to use HTTP. Other vendors have retained their original architectures, and provided a gateway server for Web-enabling their products — Microsoft Outlook, for example, includes an optional Web gateway called Outlook Web Access (OWA).[13] Still others have released a C or Java API with functions you can call from your custom Web applications. Legacy system vendors likely do not offer any of these options. In this case, you may want to consider using the Citrix business package in your portal.

  Since Web-based applications integrate cleanly and easily into a portal, your next step after considering business packages should be to explore whether you can Web-enable your non-SAP system, by upgrading, by installing a Web gateway, or through custom development. Regardless of which option you choose, you'll integrate the Web-enabled system just like any other Web-based resource: using a generic URL iView (URL iView template in 6.0; .NET Custom iView in 5.0). I'll show you how to do this when we discuss Web content integration later in the article.

- **Use a visual development tool.** SAP EP 6.0 includes a special type of iView — called the Database JDBC iView — for rendering data directly from SQL databases, which I will discuss in more detail later in the article. In addition, SAP has announced that they will release an add-on to the Visual Composer tool called the BI Kit, which will contain connectors for backend SQL and third-party systems. For details on using the Visual Composer, see my previous article in the July/August 2004 issue.

- **Launch the system's native GUI on the user's PC.** The best approach to non-SAP integration from a usability and maintainability perspective is almost always to Web-enable and integrate. Nevertheless, there are situations in which Web-enabling is too limiting. For example, if users are very experienced at using the native GUI and have advanced processing needs, deploying small sets of data via iViews will usually not meet their needs. Alternatively, if the system includes a lot of transactions or reports that need to be individually Web-enabled via custom development, it may not be worth the investment.

  In these situations, it's appropriate to launch the vendor's native GUI either within an iView embedded in a portal page or natively on the user's desktop. Users retain full access to the functionality they need, and you gain architectural simplicity and promote portal use. Since SAP EP doesn't offer a standard way to launch non-SAPGUIs, however, you'll need some custom code to do this, which is available for download at www.SAPpro.com.[14]

Figure 5 illustrates how these four strategies — using a business package, Web-enabling and integrating, using a visual development tool, and launching the native GUI on the user's PC — apply to three common types of non-SAP systems:

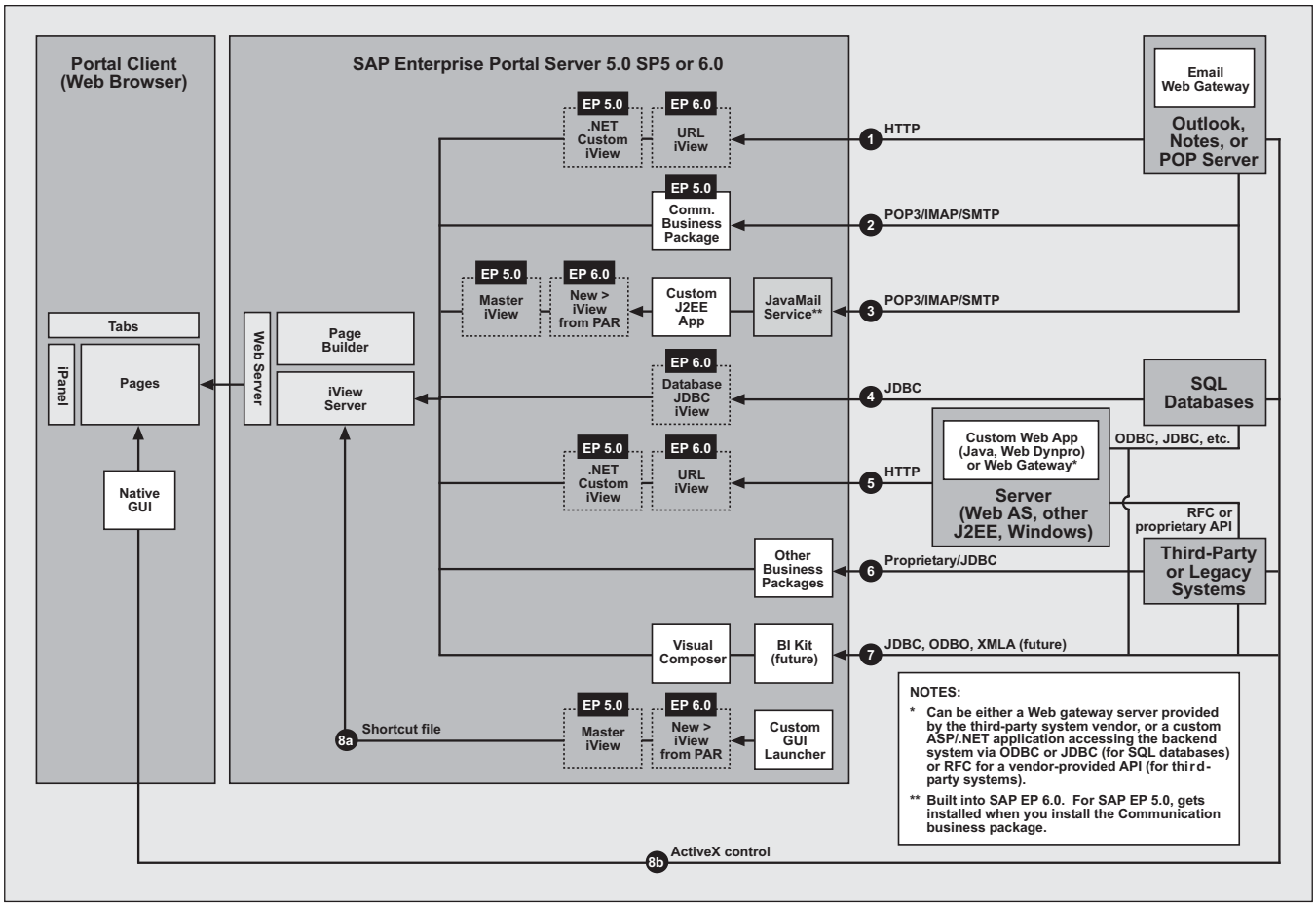
- Corporate email systems (options ❶, ❷,❸, and ❽/❽)
- Data in JDBC-compliant SQL databases (options ❹, ❺, and ❼)
- Other third-party and legacy systems (options ❺, ❻, ❼, and ❽/❽)

In the following sections, I discuss these options in detail and point out the pros and cons of each. Consider each in the context of your own requirements, current systems and releases, and upgrade plans.

[13] ITS is another example. ITS was released by SAP in the mid-1990s as a standalone server for Web-enabling SAP systems, and was available to SAP customers for free.

[14] We'll use the same approach used by SAP Transaction iViews to launch WinGUI or JavaGUI (this was covered in detail in my previous article).

Figure 5 Options for Integrating Content from Non-SAP Systems

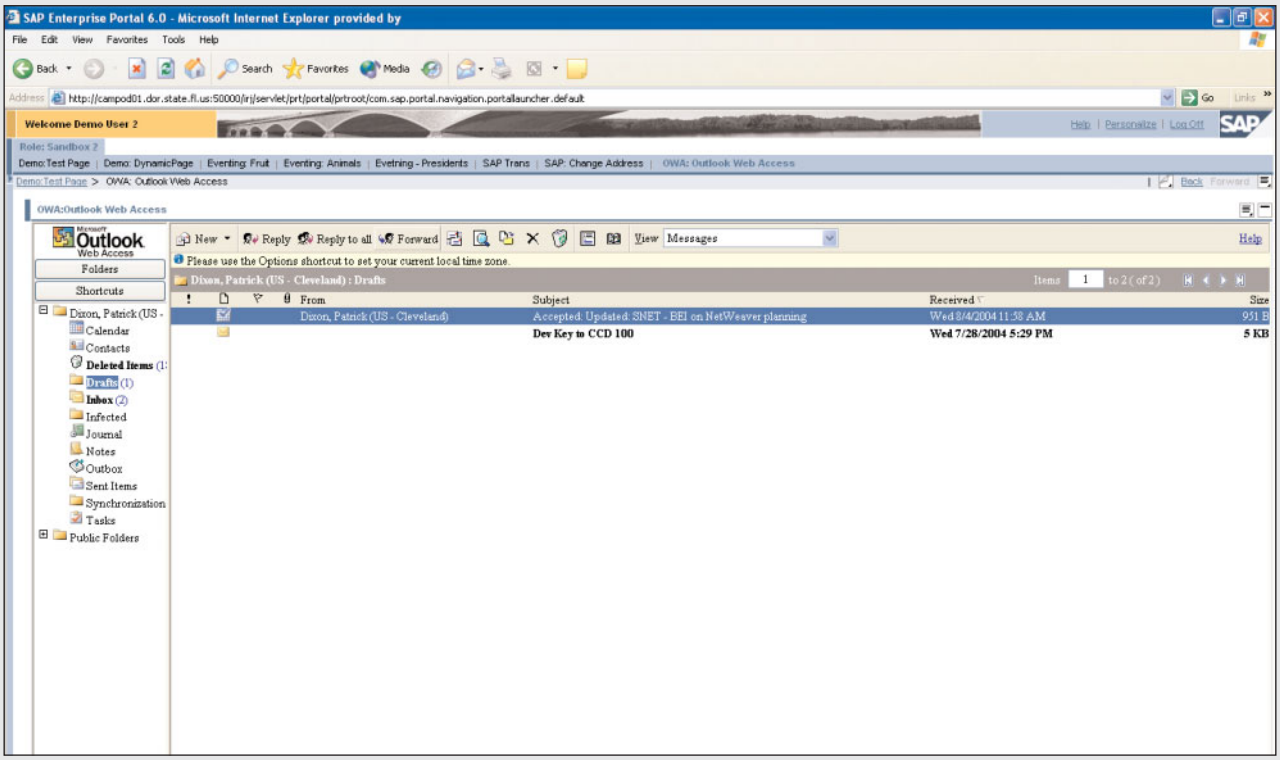


> ✓ ***Note!***
>
> *To save time and avoid delay when defining iViews, it's a good idea to set up and test your system landscape files before performing any of the techniques described in this article. Instructions on how to do this are available at www.SAPpro.com.*

Integrating Corporate Email Systems

After SAP content, the next most popular integration requirement is email. If you're planning to deploy an enterprise portal (vs. a point solution), you should strongly consider integrating your corporate email system, for two reasons. First, it helps reinforce your portal as a central workplace, even if you continue to support regular mail clients on employee PCs in parallel. Second, it makes accessing email *much* more convenient — users can instantly access their email from any PC while away from their desks, or when away from the office (if the portal is Internet-accessible). This is extremely useful for employees who work in large plants and routinely have a 15-minute or longer

Figure 6 Microsoft Outlook Web Access (OWA) Integrated As a URL iView



walk back to their desks, for example, or salespeople who would otherwise not be able to access their email from customer sites.

As mentioned previously, you have four options depending on your portal release:

- Web-enable your email system and integrate it as an SAP EP 6.0 URL iView or an SAP EP 5.0 .NET Custom iView (option ❶).
- Leverage the iViews in the Communication business package (option ❷).
- Develop your own custom iViews using the JavaMail service (option ❸).
- Integrate your email system's frontend client directly (option ❽/❽).

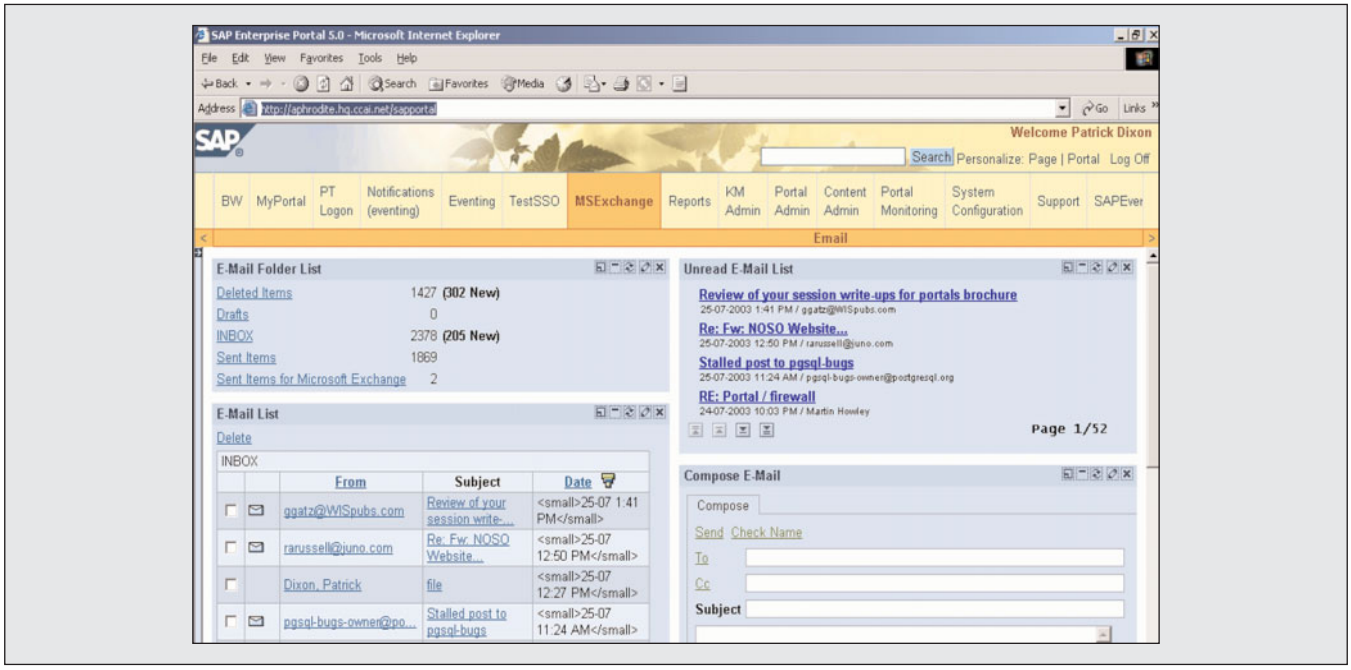
Let's discuss each option in order of its utility.

Web-Enable and Integrate Your Email System

With the release of SAP EP 6.0, SAP changed their strategy for email and calendar integration — they now recommend that SAP EP 6.0 customers use their email system's native "WebMail" interface rather than the iViews within the Communication business package. For Microsoft Outlook this is the Outlook Web Access (OWA) component delivered with Outlook 5.5 SP4 and higher (see **Figure 6**), and for Lotus Notes this is the iNotes component delivered with Notes 5.0.8 and higher. Companies using older or non-mainstream email products should strongly consider upgrading or migrating since WebMail is an increasingly important capability. Both of these applications can be integrated into the portal as generic URL iViews (see option ❶ in Figure 5), just like any other Web content.

In my opinion, using a native WebMail interface is almost always the best approach anyway (even for

Figure 7 Email iViews from the Communication Business Package



SAP EP 5.0 customers), since it offers users a familiar interface, and nearly all of the functionality of the underlying product. In contrast, the iViews in the Communication business package offer only the "least common denominator" of functionality, since they were designed to work with several different email systems, including Lotus Notes, Microsoft Outlook, and other POP/IMAP systems. So, in short, I recommend that *all* SAP EP customers use WebMail whenever possible, not just those using SAP EP 6.0.

When integrating your WebMail interface, keep three things in mind:

1. Remember to customize your WebMail application to look similar to your portal (e.g., customize the colors to match your portal's theme). If done well, this will make your portal look more professional and trick users into thinking it's a native portal application.[15]

2. Integrate your WebMail interface as its own portal page instead of making it part of one that is cluttered with other iViews.

3. Rather than force users to open the full WebMail page to check for incoming messages, provide a very small iView on the main portal page that lists the number of unread messages in their inbox, for example. There are two ways to get this type of iView: either install and "steal" it from the Email Folder List iView in the Communication business package, or develop your own as a JavaServer Page (JSP) that leverages the JavaMail service built into SAP EP 6.0 (more on this in an upcoming section). The Communication package, which we'll look at next, can technically be installed on either SAP EP 5.0 or 6.0, although officially it is only supported for 5.0.

Leverage the Communication Business Package

The Communication business package (see option ❷ in Figure 5) contains out-of-the-box email iViews for Microsoft Exchange, Lotus Notes, and POP/IMAP-compatible systems (see **Figure 7**). The package also includes specialized iViews for Microsoft Exchange

[15] The colors for Outlook Web Access, for example, can be customized by manually modifying users' underlying .ASP files (OWA 5.5) or by defining themes in OWA 2003.

Figure 8 Connecting to a Microsoft Exchange Server to Integrate Email iViews

```
<System name="SAPPortals_JavaMailConnector">
<!--Specify "Microsoft_Exchange_Server", "Lotus_Domino_Server",
"COMAIL_SERVER" below, depending on the type of email system you are trying
to connect to-->
<pcd:Attribute name="MailServer" value="Microsoft_Exchange_Server"/>

<!--Specify the network location of your outgoing SMTP server-->
<pcd:Attribute name="OutgoingSMTPServer" value="MyMailServer.com"/>
```

and Lotus Notes for calendaring and address book functionality. The iViews are generic and access a subset of the functionality in each package. Consider using this package only when you can't (or don't want to) set up a WebMail interface, or when you just want to try to use one or two iViews in addition to your WebMail, such as the unread messages or calendaring iViews. As mentioned in the previous section, while the Communication package is only officially released and supported for SAP EP 5.0 (see SAP Note 642775), technically it can be installed on both SAP EP 5.0 and 6.0 systems.

> **Caution!**
>
> *Especially if you are running SAP EP 6.0, install and evaluate the Communication package on a sandbox system first in order to detect any adverse consequences. I have done it without incident, but it pays to be cautious since business packages cannot be uninstalled!*[16]

Like all business packages, the Communication package is easy to install — just download the package to a temporary directory on your portal server, and import it from within the Web-based portal administration tool (*Portal Admin → Import*). For email integration, in SAP EP 5.0 all you need to do is insert a few entries into the systems.xml file — e.g., the email server location, whether to connect via IMAP or POP3, etc. To make this easier, the installer creates a template file containing example XML entries in the J2EE "irj" subdirectory. The best approach is to paste entries from this template directly into your systems.xml file and make any required modifications. **Figure 8** shows some example entries for connecting to a Microsoft Exchange server. In SAP EP 6.0, you use a visual administration tool to configure the portalapp.xml file.[17] For details, go to http://help.sap.com/portals and navigate to

> **Note!**
>
> *If you plan to use the Communication package's calendaring and address book iViews, be aware that you'll need to install additional files (like .DLLs) on your Exchange or Notes servers, and that you'll also need a Microsoft IIS to run the .ASP files underlying the Exchange address book and calendaring iViews. Avoid surprises by reviewing the specific instructions and installation requirements in the documentation before downloading and installing the package.*

[16] You have to identify and delete the package components — i.e., iViews, pages, roles — individually.

[17] In SAP EP 5.0, system definitions are stored in two files — systems.xml and JCODestinations.xml. In SAP EP 6.0, system definitions are stored in a single file called portalapp.xml (see the sidebar on page 91 for more details).

SAP Enterprise Portal Documentation → Administration Guide → Portal Platform → System Administration → System Landscape → System Landscape Editor (see also the download available at www.SAPpro.com).

Overall, the installation is straightforward. Just follow the package documentation available at the Portal Content Portfolio site.

Develop Your Own Custom iViews Using the JavaMail Service (SAP EP 6.0)

The email iViews within the Communication business package use the JavaMail service to connect to back-end systems. Technically, the JavaMail service is just an API for accessing SMTP, POP3, and IMAP4 mail servers, and for transporting, storing, and accessing a message on the Internet or on your internal network. Since the JavaMail service complies with Sun's J2EE JavaMail 1.2 specification, you can use it in any of your custom Java applications that are within the portal's J2EE runtime. This also means that sample code and documentation are widely available.

To build your own iViews based on the JavaMail service (see option ❸ in Figure 5), you need to develop and deploy a custom portal component. If you have SAP EP 5.0, you do this by downloading IBM's Eclipse development environment from www.eclipse.org and the 5.0 version of SAP's Portal Development Kit (PDK) from http://sdn.sap.com.[18] The PDK 5.0 includes a PC-based test environment for testing your components, as well as documentation on the portal's public Java API. If you have SAP EP 6.0, you'll need to download SAP NetWeaver Developer Studio and the 6.0 PDK from http://sdn.sap.com. SAP NetWeaver Developer Studio is built on Eclipse, and includes a development environment with portal-specific development options.[19] The 6.0 PDK is released as a business package that installs example code and portal component testing tools right into your portal administration tool (log on to the portal administration tool and access the Java tab).

While developing custom components is beyond the scope of this article, **Figure 9** shows some simple Java code that retrieves the number of unread messages from a POP3-compliant email server.[20] Notice how the Java mail components are stored in the `javax.mail` package. You can leverage the `GetUnreadMessagesCount` class within your custom portal components (servlets).

[18] See the article "PDK Installation and Customization for SSO Access to SAP Systems: Essential Lessons for Developers and Implementation Teams" (*SAP Professional Journal*, November/December 2002).

[19] See the article "Get Started Developing, Debugging, and Deploying Custom J2EE Applications Quickly and Easily with SAP NetWeaver Developer Studio" (*SAP Professional Journal*, May/June 2004).

[20] Most email servers are POP3-compliant.

Figure 9 Example Java Code to Count Unread Messages from a POP3 Email Server

```
import javax.mail.*;
import javax.mail.internet.*;

public class JavaMailExample {

    public static int GetUnreadMessagesCount(String host, username, password)
   {
        // Create a new session
      Session session = Session.getInstance(
            System.getProperties(), null);

      //  Retrieve a references to the email store
      Store store = session.getStore("pop3");
```

(continued on next page)

Figure 9 (continued)

```
        store.connect(host, username, password);

        // Retrieve a read-only reference to the inbox folder
        Folder folder = store.getFolder("INBOX");
        folder.open(Folder.READ_ONLY);

        // Compute the number of unread messages
        Int count = folder.getUnreadMessageCount();

        // Close the connection
        folder.close(false);
        store.close();

        return count;
    }

    public static void main (String args[])
    {
        String host = args[0];
        String username = args[1];
        String password = args[2];

              System.out.println("Unread messages waiting: " +
GetUnreadMessagesCount(host, username, password));
    }
}
```

> **Caution!**
>
> *Note that the JavaMail service does not provide access to calendars, address books, tasks, or other functionalities built into many email systems, so if you need access to these items, you'll need to either look for vendor-provided Web-enablement for the item, or draw iViews from the Communication business package.*

> **Note!**
>
> *For a quick overview of the JavaMail API and to see additional programming examples, visit www.javaworld.com/javaworld/jw-06-1999/jw-06-javamail.html. For details on the JavaMail API, visit http://java.sun.com/products/javamail/ and http://java.sun.com/products/javamail/javadocs/index.html.*

Integrate Your Email System's Frontend Client Directly

If the previous options are too costly, too complicated, or don't provide users enough functionality, there is another option: launch your email application's native GUI on user PCs. I'll show you two ways to do this (option ⓑ/ⓒ in Figure 5) for all third-party GUIs (not just email GUIs) later on in the article. If you choose this option, I strongly recommend that you launch the

client natively via Windows instead of embedding the application within an iView, to avoid confusion and to make the best use of screen real estate. I also strongly recommend you add a "number of unread messages" iView to the user home page, as discussed previously, so users don't have to launch the full GUI just to check if new mail has arrived. Both will greatly enhance user satisfaction and portal adoption.

Integrating Data in JDBC-Compliant SQL Databases

SQL database integration is an increasingly common requirement on portal projects, as most companies have existing SQL data that users can benefit from, or want to protect their SAP systems from "untrusted" data. One company I consulted with, for example, wanted external users to be able to update their address information, but wanted the details to be validated by administrators before updating the SAP database. They chose to store user updates in a separate SQL database created on the same Oracle instance as their SAP database.

> ✓ ***Note!***
>
> *Before you can create JDBC iViews or access SQL databases from your custom applications, you'll need to install one or more drivers on your portal server(s). The drivers you'll need depends on both the database and the method by which you'll access it (i.e., ODBC or JDBC).*[21] *If you have SAP EP 6.0, you'll only need to consider JDBC drivers, since ODBC and ASP/.NET technologies are no longer supported. If you have SAP EP 5.0, you'll want to install both the ODBC and JDBC drivers for each of your databases, for maximum flexibility. Many of my clients have found the documentation confusing, so I've included clearer instructions in a document available for download from www.SAPpro.com.*

[21] ODBC is a Microsoft technology for accessing heterogeneous SQL databases in a common way from Visual Basic, ASP/.NET, etc. JDBC is its complement on the Java side.

As shown in Figure 5, there are three main options for integrating SQL data:

- Create Database JDBC iViews to automatically render SQL data (option ❹).
- Leverage connectors in the Visual Composer BI Kit to visually design iViews (option ❼).
- Develop a custom Web application to access your SQL data via JDBC or ODBC (option ❺).

We'll discuss each option in turn in the following sections.

Create Database JDBC iViews

SAP EP 6.0 includes a new type of iView called the Database JDBC iView that operates similar to the SAP Connector iView (covered in detail in my previous article) and automatically generates a rudimentary HTML interface for data within the database. You define the iView visually with a wizard, and choose one of two data retrieval options: (1) run a stored procedure within the database, or (2) execute an SQL statement that you specify at design time. The second option offers a very fast and inexpensive way to provide limited access to SQL data. Most customers find it more useful as a testing tool than a productive tool, however, due to its display limitations — e.g., the displayed field labels are usually nonintuitive technical field names, and are not customizable.

> ✓ ***Note!***
>
> *In contrast to the SAP Connector iView, Database JDBC iViews are display-only, meaning that you cannot change data in the database.*

Figure 10 Definition and Output of a Database JDBC iView

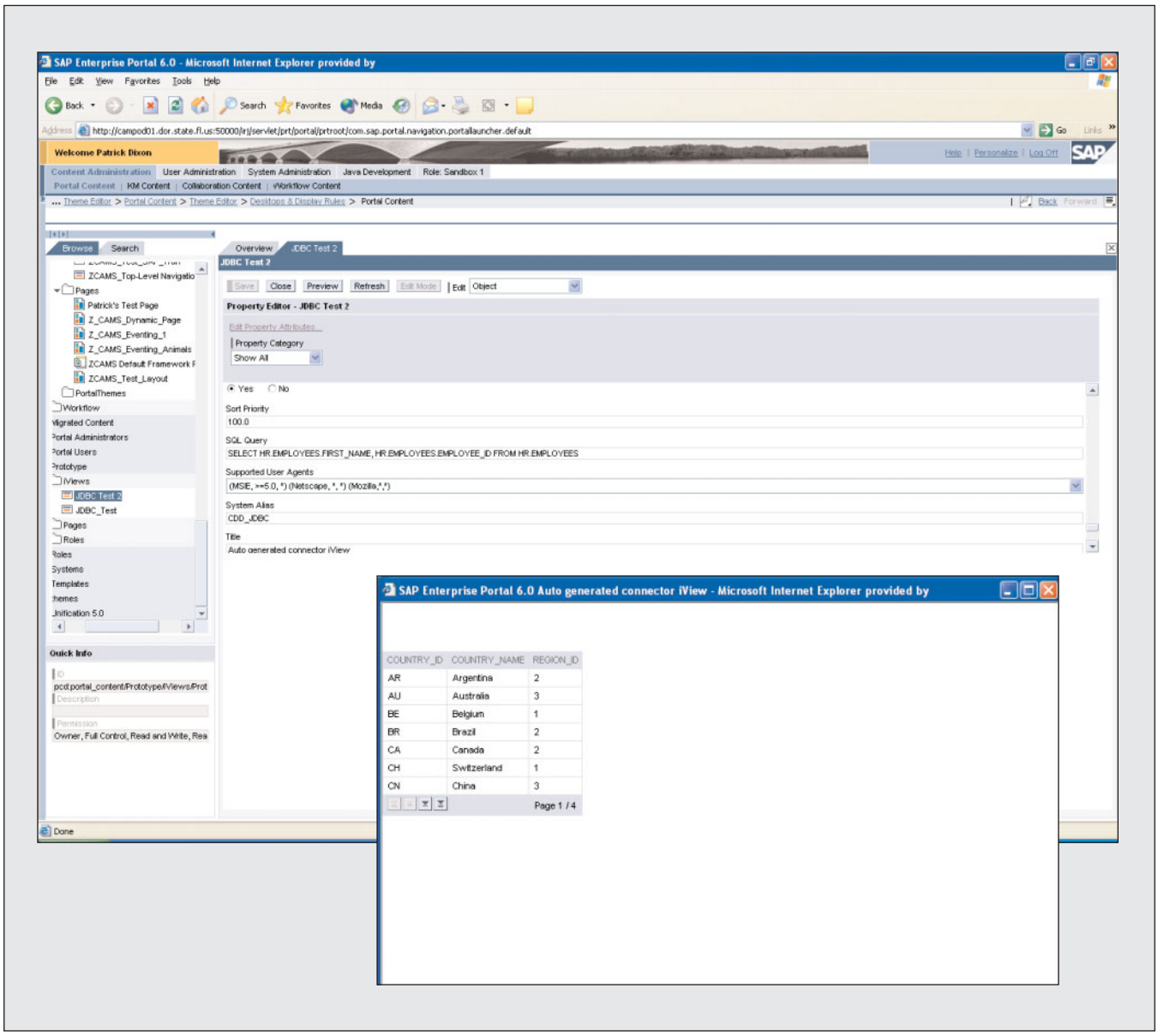


Figure 10 shows the definition and output of a sample Database JDBC iView (option ❹ in Figure 5). Once you've defined the connection to your SQL databases in the System Landscape Editor, as described previously (see also the download at www.SAPpro.com and the SAP online help), they will appear as systems in the dropdown list in the SAP connector of the iView wizard, and will be available for editing in the System Alias box shown in Figure 10.

Leverage Connectors in the Visual Composer BI Kit

Although details are forthcoming, SAP has announced plans to release a set of non-SAP connectors for the Visual Composer tool as part of an add-on called the BI Kit. Included in these will be a JDBC connector for connecting to backend SQL databases, so I've included this as an option in Figure 5 (option ❼) for

completeness.[22] As of this writing, the Visual Composer is available as an add-on for SAP EP 5.0, and an SAP EP 6.0 version will soon be released publicly.

> ✓ ***Note!***
>
> *As discussed in my previous article, the Visual Composer has two parts: a storyboard development component, and a portal add-on component with libraries needed to run iViews developed with the Visual Composer. Remember to only install the add-on libraries on your production portal, and to build and transport your Visual Composer iViews on a development system using the export and import features in the portal administration tool. Also keep in mind that Visual Composer requires a Microsoft SQL Server.*

Once available, this option will offer you a flexible way to build iViews. The Visual Composer goes beyond Database JDBC iViews by allowing input fields, control of field placement, and customizable field names. Periodically check the SAP Enterprise Portal area of http://sdn.sap.com[23] for more information and the latest on add-on availability.

[22] Current plans for the BI Kit also include connectors for OLE DB for OLAP (ODBO), XML for Analysis (XMLA), and SAP Query.

[23] Go to http://sdn.sap.com and select "Enterprise Portal" from the Developer Areas navigation bar.

Write a Custom Web Application Using ODBC or JDBC

If the Database JDBC or Visual Composer is insufficient for your needs — if you need to merge data from more than one database, for example — you're next best option is to develop a custom Web application using Java or Microsoft ASPs, and integrate it as a generic URL iView (option ❺ in Figure 5). As mentioned previously, as of SAP EP 6.0 the portal supports only Java natively, but you can conceivably host ASPs on your portal server by running a Microsoft IIS Web server in parallel (if your portal server runs on Microsoft Windows). Another approach is to host ASPs on a separate Windows server, although the performance will degrade somewhat due to the additional network communication.

If you choose to use Java, you can either develop your Web application as a traditional JSP or servlet, or you can build a Web Dynpro program with SAP NetWeaver Developer Studio, which is available for download from http://sdn.sap.com. SAP NetWeaver Developer Studio includes portal-specific libraries you can use to access portal runtime variables, tailor your output to the current portal theme, etc. You'll find detailed examples on how to do this, and documentation on the Java API provided by the SAP EP runtime, in the PDK, which is also available for download from http://sdn.sap.com.

To give you an example of what's involved in this approach, I've included code in **Figure 11** from

Figure 11 Java Code to Access Data in an SQL Database via the Portal Server JDBC Driver

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;

public class DemoJDBCServlet extends HttpServlet
{
      public void doGet (HttpServletRequest req, HttpServletResponse res)
          throws ServletException, IOException
      {
          res.setContentType("text/html");
```

(continued on next page)

Figure 11 (continued)

```
          ServletOutputStream out = res.getOutputStream();
          out.println("<html>");
          out.println("<head><title>Hello World</title></head>");
          out.println("<body>");

            try
            {
            Class.forName ("oracle.jdbc.driver.OracleDriver");
            }
            catch (ClassNotFoundException e)
            {
            out.println("Unable to load driver.");
              e.printStackTrace();
            }

            try
            {
                  Connection conn = DriverManager.getConnection
                  ("jdbc:oracle:oci8:@hostname_orcl", "scott", "tiger");

              Statement stmt = conn.createStatement();
              ResultSet rset = stmt.executeQuery("select BANNER from
SYS.V_$VERSION");
                  while (rset.next())
                out.println (rset.getString(1));   // Print col 1
                  rset.close();
                  stmt.close();
                  conn.close();
            }
          catch (SQLException e)
            {
                  out.println("SQLException: " + e.getMessage());
                  out.println("SQLState:     " + e.getSQLState());
                  out.println("VendorError:  " + e.getErrorCode());
            }

      }
}
```

a sample servlet that connects to a backend database using the built-in JDBC driver within the portal server. You can build a portal component based on this code within SAP NetWeaver Developer Studio. The tool will output a PAR file that you can import into your portal by selecting Based on Master iView in the Java iView Editor dialog (see Figure 3) in SAP EP 5.0, or by right-clicking on the target folder in the PCD and selecting *New → iView* (see Figure 4) in SAP EP 6.0.

So far, we've covered integration techniques for

email systems and SQL databases. We'll next discuss techniques for third-party and legacy systems, including content management systems like Documentum and IXOS, external accounting or production systems, and so on.

Integrating Other Third-Party and Legacy Systems

As shown in Figure 5, you have four options for integrating other third-party and legacy systems:

- Option ❺ involves Web-enabling your system either through a vendor-provided gateway[24] or through a custom application that leverages a vendor-provided API library.
- Option ❻ involves leveraging any Portal Content Portfolio business packages provided by SAP or a vendor.
- Option ❼ calls for you to keep a lookout for new connectors to the Visual Composer that could be used to connect to your non-SAP system.
- Option ❽/❽ involves launching third-party GUIs on user PCs.

> ✓ ***Tip***
>
> *API libraries for third-party systems are usually available for download at the vendor's Web site as part of a complete Software Developer's Kit that usually includes code samples. Some vendors only provide them upon customer request, so make sure to ask if you don't find any online.*

We discussed and applied options ❺, ❻, and ❼ earlier in this article, so we need not discuss them again. Option ❽/❽ warrants additional discussion since SAP EP doesn't provide an out-of-the-box solution for launching third-party GUIs from portal pages. This isn't an oversight, however; it results from an important technical restriction: browsers like Microsoft Internet Explorer intentionally restrict Web page objects from launching programs on a user's PC. Without this restriction, a hacker could embed some malicious code within a Web page to delete all files from your C drive using the Windows shell program, for example.

There are two ways you can overcome this limitation. The first, more elegant approach is to develop (or purchase) a simple, generic ActiveX control that can launch a program on a user's PC (option ❽ in Figure 5). Unlike JavaScript, VBScript, or even Java applets, ActiveX controls are granted nearly unlimited access to the Windows API, and can therefore launch programs, create or access files, etc., just as if it were a Visual Basic executable running on the user's computer. The only security standing in the way is specific browser security settings that govern when ActiveX controls can be downloaded and run. You can view these settings in Internet Explorer by navigating to *Tools* → *Options* → *Security*.

> ✓ ***Tip***
>
> *Particle Software sells an ActiveX control called IntraLaunch that can be used for this purpose. For an online demo, go to the Particle Web site (www.particlesoftware.com), click on the online demo link, and select "Corporate Portal" from the dropdown menu. (Note that I am not compensated by or affiliated in any way with this vendor.)*

If you choose this solution, there are a few things to keep in mind, however:

- The default security settings for ActiveX controls differ according to Internet Explorer version, and whether the controls are being downloaded from an intranet location or from the Internet. If you

[24] A gateway that either is a part of or resides on the third-party server itself, or is set up on an external server.

will have both intranet and Internet users, make sure you anticipate what their settings will be.

- There are different security settings for *signed* ActiveX controls and *unsigned* ActiveX controls. Controls are unsigned by default, and can be signed with a special tool from Microsoft that needs a private key purchased from a certification authority (CA) like VeriSign to guarantee your identity. The default for Internet Explorer 6.0 for both Internet and intranet zones is to *disallow* downloads of unsigned controls, so signing your control (or purchasing a signed control), instead of modifying the browser settings for each and every user, will greatly simplify deployment.

- Internet Explorer downloads and stores ActiveX controls for future use, even if the user clears the browser history or the temporary Internet file cache. This has three important effects. First, when users initially access the page containing the control, the page will load quite slowly, since the ActiveX control is downloaded when the page is loaded. Be sure to alert your support staff in case they get complaints. Thereafter, the browser will recognize that the control is already on the PC. Second, in most cases the user will receive a single warning dialog (that asks permission to download and run the control) just before this one-time download; thereafter the control will function without interaction. Third, be aware that Internet-only users (like customers or vendors) may report different issues than internal users who take their PCs home and access the portal over the Internet, since internal users may have already downloaded the control while on the intranet.

> ✓ ***Note!***
>
> *If your users use browsers other than Internet Explorer — e.g., Netscape Navigator — they'll need to install an ActiveX plug-in (available at www.microsoft.com) to be able to run the control.*

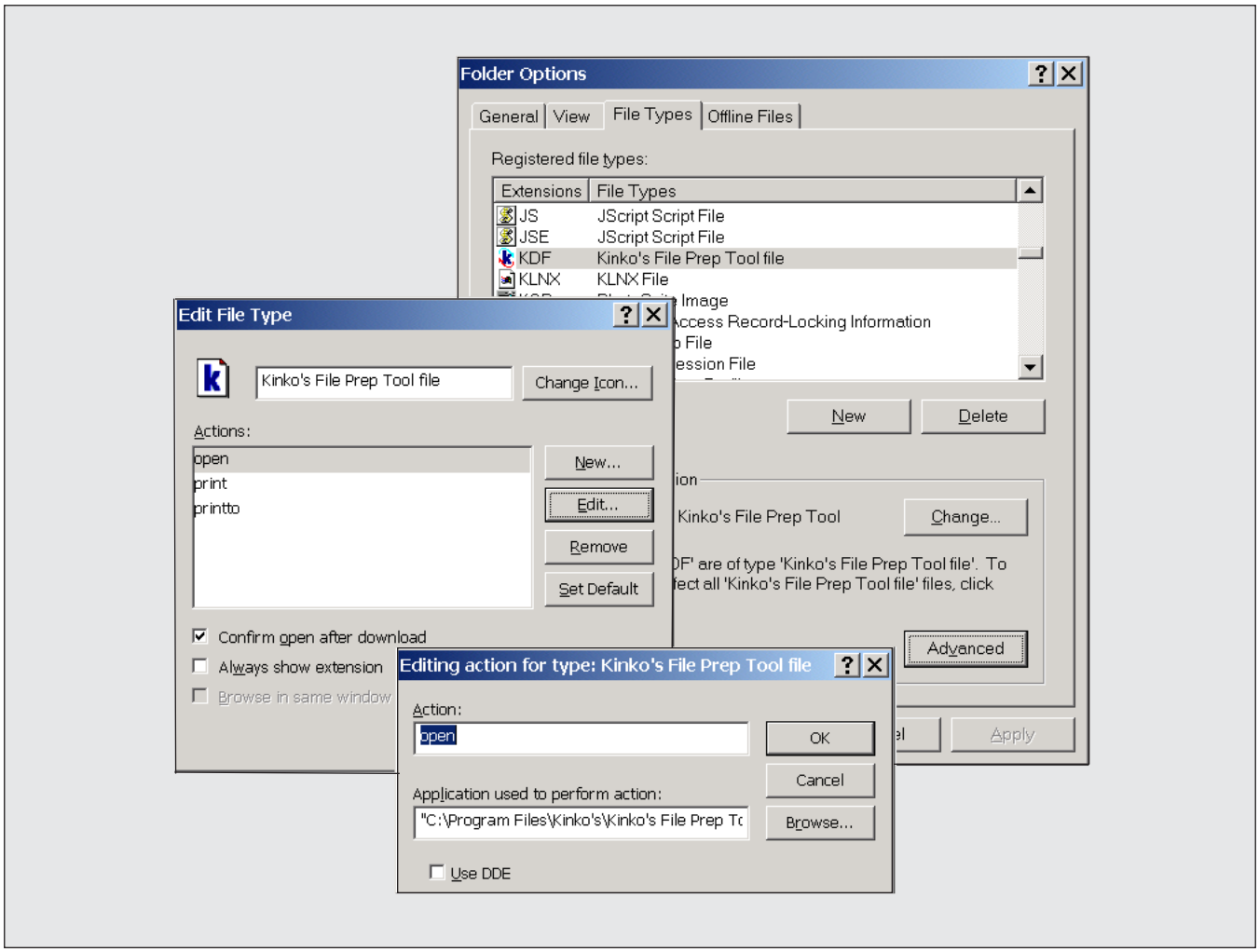
If you find this solution too expensive or too complex, there is an alternative but imperfect way to launch third-party programs: you can create a hyperlink to a Web application that dynamically generates a dummy "shortcut" file that is mapped to the third-party GUI program you want to launch (option ⓒ in Figure 5). Sound complicated? It isn't, really. In my previous article, I demonstrated how to have your portal launch SAPGUI for Windows natively on a user's desktop, instead of embedded within a browser window. I explained that behind the scenes, the portal generates a shortcut file that the browser downloads and passes to Microsoft Windows to open. SAP shortcut files have the extension .SAP and contain the name of the transaction that SAPGUI should run. When you install SAPGUI on a PC, the installer registers the .SAP file type in Windows and specifies that Internet Explorer should not bother the user with a download confirmation prompt when the browser receives an incoming file with the .SAP extension. To the user, all of this happens seamlessly; they just click a link to access the portal page, which contains the iView defined to launch the transaction with WinGUI, and up it pops.

To emulate this for third-party GUIs, I wrote a simple .ASP file called shortcut.asp — downloadable from www.SAPpro.com — that accepts a URL parameter called "extension" and outputs a blank shortcut file called shortcut.extension with an "application/extension" MIME type. All you need to do in order to use it is the following:

1. Place the shortcut.asp file somewhere on a Microsoft IIS Web server, and note the URL required to access it — e.g., http://www.myco.com/aspfiles/shortcut.asp. In practice, you'll specify the desired file type extension in the URL, such as http://www.myco.com/aspfiles/shortcut.asp?extension=KDF.

2. Next, investigate whether there is an existing file extension on user PCs that can be used to launch the GUI program. To do this, go to Microsoft Windows Explorer, and follow the menu path

Figure 12 Creating a Shortcut That Automatically Launches a Third-Party GUI



✓ ***Important!***

To prevent users from getting a file download confirmation dialog each time their browser receives the shortcut file, you'll need to uncheck the "Confirm open after download" checkbox (see the Edit File Type dialog in Figure 12). This can be a challenge if you have thousands of user PCs, so ask your Windows administrator if there is an automated desktop maintenance tool that can do this en masse.

Tools → Folder Options → File Types (see **Figure 12**). If SAPGUI is installed, you should see the extension .SAP in the list. To see which program is bound to a particular file extension, highlight the item, click on the Advanced button, select the "open" action, and click on the Edit button. If you find a file type that will work, you can skip to step 4.

3. If you can't find a file type that will launch the program, you can create your own! Just click on the New button in the Folder Options dialog, and enter a three-letter file extension. Then select your new type from the file types list, click on the Advanced button, and click on the New button to define a new action. Specify "open" for the action name and a path to your GUI's executable in the "Application used to perform action" field. Finally, click on OK and uncheck the "Confirm open after download" checkbox. Again, ask your Windows administrator how to do this en masse on user PCs.

4. Finally, place a hyperlink to generate the shortcut either on a portal page or in a standalone URL iView. For example, if we were to define a custom extension ZZX, the link would be http://www.myco.com/aspfiles/extension=ZZX.

The result is that the GUI program should pop up without a dialog if the "Confirm open after download" option is unchecked. Otherwise, the user will receive the standard download dialog with open or run options, which can be quite confusing.

Remember that you can also use the techniques described here to launch your email application's native GUI on user PCs.

You now know how to integrate your corporate email system, your JDBC-compliant SQL databases, and any other third-party and legacy systems into your portal. Next, and finally, we'll look at how to integrate Web-based content.

Integrating Existing Web-Based Content

Perhaps the most important integration feature of SAP EP is its ability to bring in any type of Web content. For this reason, the portal almost always replaces a company's traditional "intranet" — employees use the portal to access everything, regardless of where it is hosted. In this section, we'll explore your options for integrating Web content (see **Figure 13**), which include:

- Leverage business packages for Web-based content, provided primarily by Internet vendors (option ❶)
- Link URLs with the generic URL wizard (option ❷)
- Use the SAP EP 5.0 iView Catcher to extract portions of Web pages (options ❸)
- Develop your own XML parser to extract portions of Web pages (option ❹)

I'll cover each option below in order of usefulness.

> ✓ ***Tip***
>
> *When planning your portal rollout to a new user group, investigate which intranet content and applications they currently access. Including all of these in the initial cutover — even if they don't share the portal's standardized look and feel — is an important element to promote portal use, and avoids political resistance from users and developers of those applications. Over time, as your portal becomes the de facto access point for intranet content, it will be much easier to convince those other groups to port their content/applications to use the new portal look and feel.*

Leverage Business Packages for Web-Based Content

If you've searched the Internet recently, you'll undoubtedly agree that it has become increasing difficult to find information that is both concise and relevant to your needs. Even with great search engines like Google, searching can be both time consuming and frustrating. Increasingly, corporations are turning to subscription-based information services, including Yahoo!, LexisNexis, and YellowBrix.

Figure 13 Summary of Web Content Integration Options

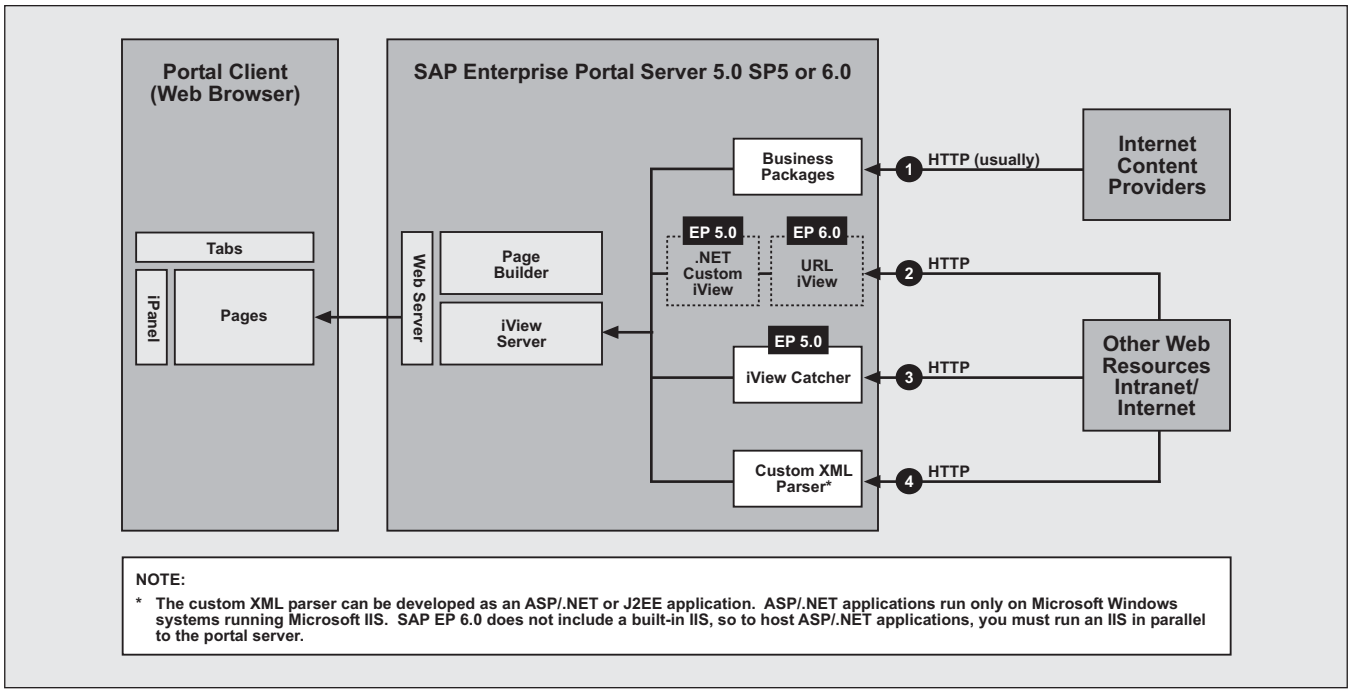
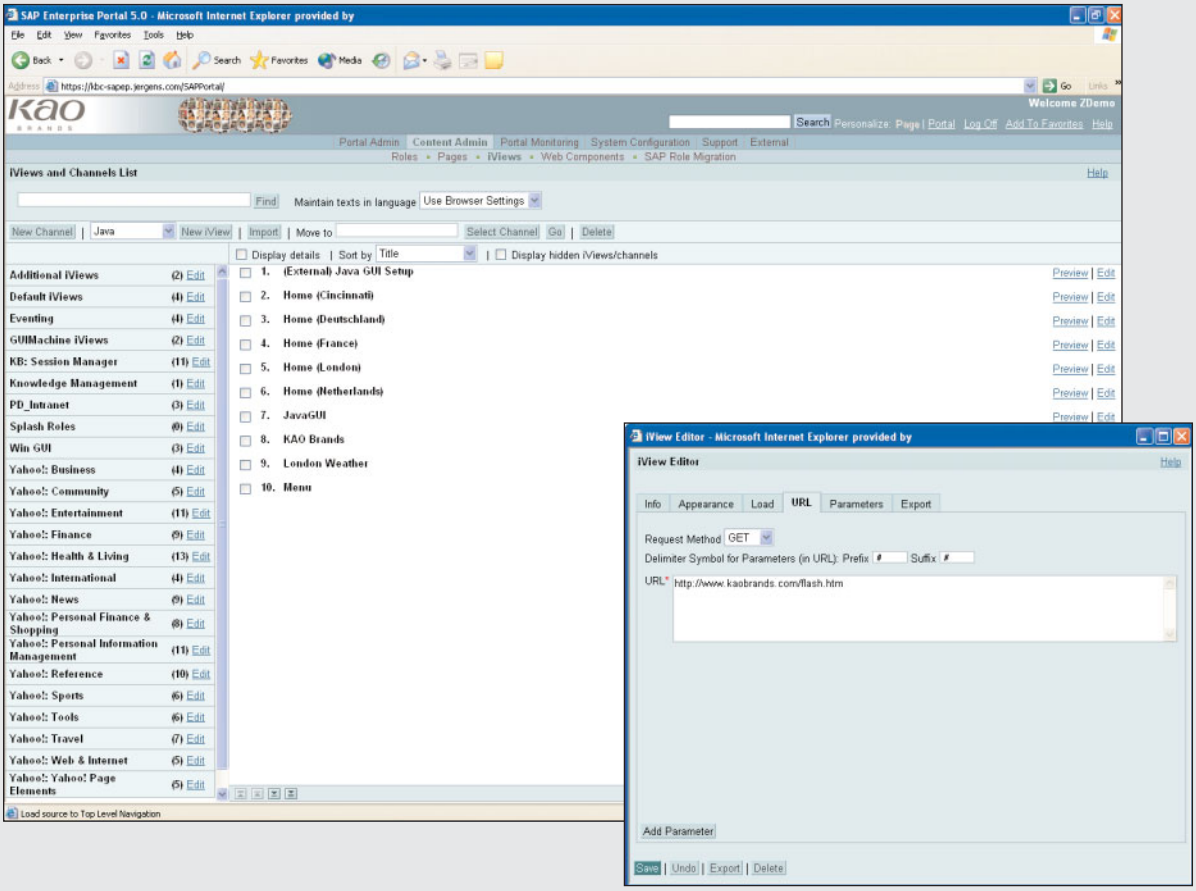


Figure 14 Example iViews in the getAbstract Business Package

Many of these information providers offer business packages in the Portal Content Portfolio (option ❶ in Figure 13) with custom iViews that deliver targeted business information, sometimes specific to a user's region, industry, or category preferences. An example is the getAbstract business package (see **Figure 14**), which provides abstract summaries of business books that might be of interest

Figure 15 Creating a URL (.NET Custom) iView in SAP Enterprise Portal 5.0



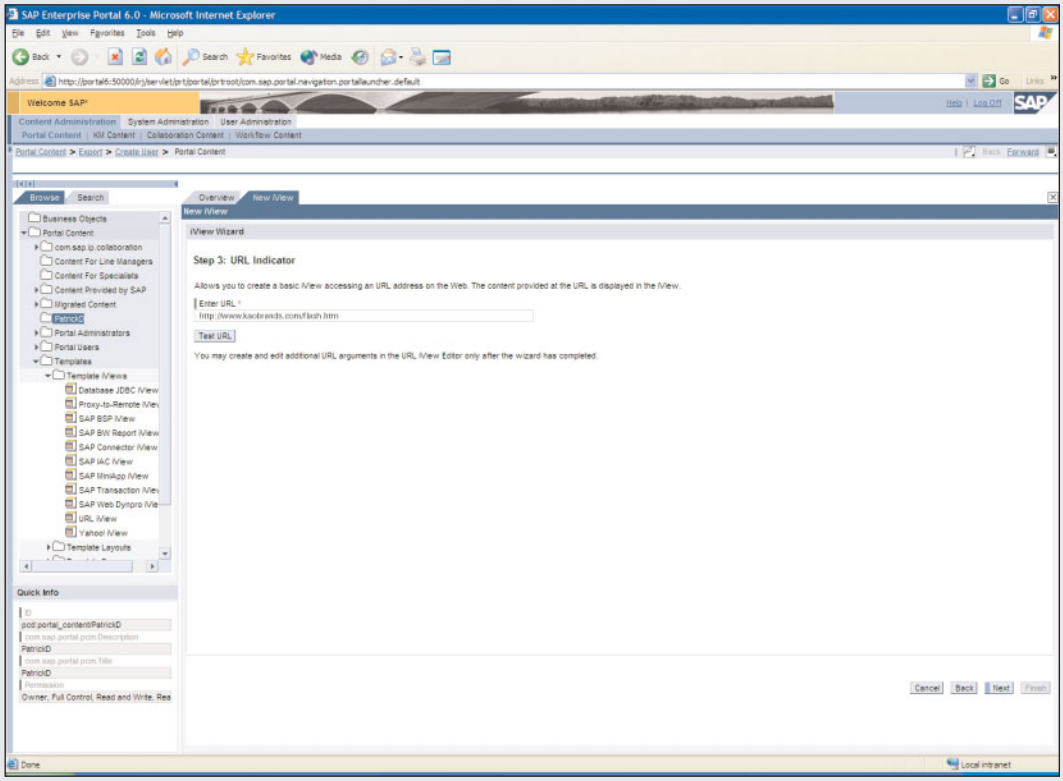
> ✓ ***Note!***
>
> *While you'll see a business package from Yahoo! in the Portal Content Portfolio, the partnership between SAP and Yahoo! was terminated in early 2004, and new licenses are no longer being offered. Yahoo! offered a corporate interface to their Internet site (nicknamed "Yahoo without ads"). For up-to-date details see http://help.sap.com/portals.*

to business readers. The iViews offer users flexible options for controlling which abstracts appear, from a general "Abstract of the Week via keyword" search, to a role-based search. The best way to investigate these iViews is to log on to the Portal Content Portfolio site and do a search.

Link URLs with the Generic URL Wizard

While I've mentioned incorporating URLs into your portal (option ❷ in Figure 13) several times in this article series, I have yet to show you how to actually do it. It's quite simple — all you need is a URL to the

Figure 16 Creating a URL iView in SAP Enterprise Portal 6.0



> ✓ ***Note!***
>
> *In SAP EP 5.0, the Portal Users business package was used to provide portal users with quick and convenient access to several basic services at a single central location, such as an overview of their tasks and work items. With SAP EP 6.0, SAP wisely decided to include these iViews in the standard portal installation. Also in SAP EP 6.0, this basic functionality has been enhanced — the portal now provides a Universal Work List iView that enables portal users to centrally manage their tasks in separate workflow systems, including SAP R/3 Business Workflow and Java Ad Hoc Workflow. Java Ad Hoc Workflow allows users to spontaneously create and track tasks through a wizard.*

Web content or application. The process is essentially the same for SAP EP 5.0 and SAP EP 6.0, although the terminology and screen appearance differ slightly:

1. Launch the iView wizard. In SAP EP 5.0, go to *Content Admin* → *iViews* and select .NET Custom iView from the dropdown list (see Figure 3). In SAP EP 6.0, right-click on the desired target folder in the PCD, select *New* → *iView*, and choose the URL iView template (see Figure 4).

2. Insert the URL in the dialog. The dialogs for SAP EP 5.0 and 6.0 are shown in **Figure 15** and **Figure 16**, respectively.

Parameterizing the URL

When developing Web applications, programmers often design application components (i.e., servlets, ASPs, etc.) to return different data depending on the information passed in by the user. For example, a "search" servlet might generate a search input form without input parameters when it is first invoked. Once the user enters values, the form could then call the same servlet, which would then detect the presence of these parameters and execute the search.

One way to pass input data to a Web program is by appending parameters (name and value pairs) to the URL used to invoke it. There are two ways to do this, and the one you'll use depends on how the target program has been coded. The more traditional, "explicit" way is to add name and value pairs after a question mark at the end of the URL. For example:

http://www.myco.com/testpage.asp?zipcode=10527&timezone=est

This method is easy to read, and the order of the parameters is unimportant since the parameter names and values are grouped together.

A second, more compact way is to embed only the parameter values in the body of the URL. In this case, the Web program must know where to look for the values, and their order is critical. The Yahoo! weather program uses this approach. For example, the URL to get the weather in Cleveland is:

http://uk.weather.yahoo.com/USOH/USOH0195/index_c.html

whereas the URL to get the weather in Clarington is:

http://uk.weather.yahoo.com/USOH/USOH0190/index_c.html

Now suppose we want to leverage the Yahoo! weather Web program to display a weather forecast on user home pages according to the user's location. One way to do this would be to define a URL iView for each city and assign the appropriate iView to each user's home page. A better approach, however, would be to define one iView and parameterize the variable element of the URL — e.g., USOH0195. The user would then maintain his or her ZIP code in the personalization area of the portal. Simply define parameters — delimited by a pound sign (#) — in the

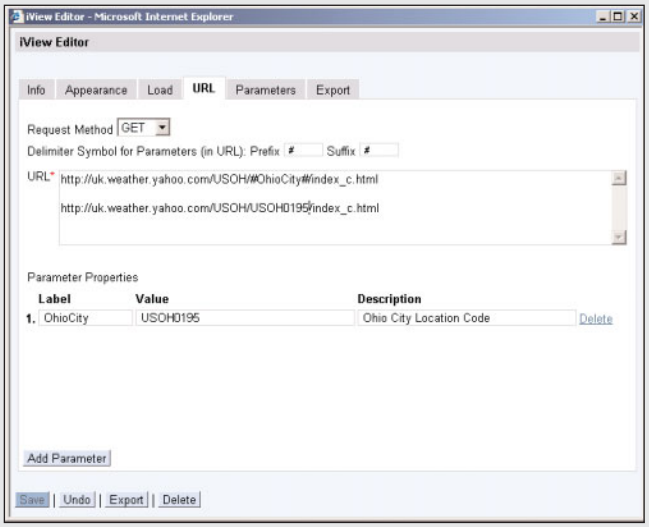
✓ ***Note!***

Parameterization is also very useful when moving between different Web environments, such as development and production, as the iView URL can be changed without having to change the underlying code.

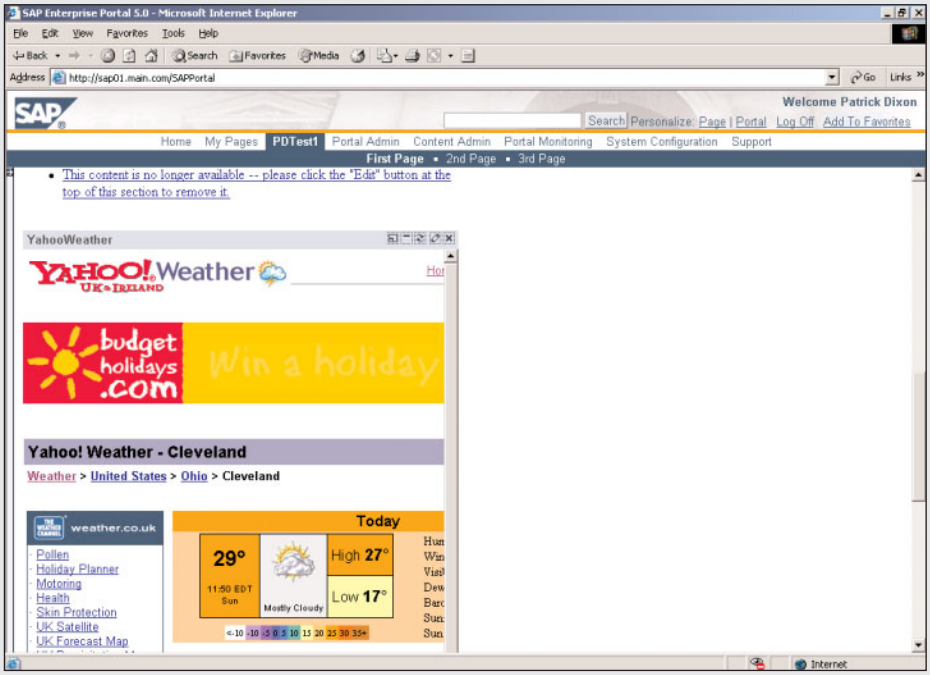
Note that you can also pass parameters to a Web program via the URL (see the sidebar above).

3. Save the iView and preview it by clicking on the Preview button.

URL tab of the iView Editor (see the screenshot to the right) and mark them as personalizable in the Parameters tab. At runtime, the user can maintain the values by clicking on the Personalize iView icon at the top right of the iView pane.

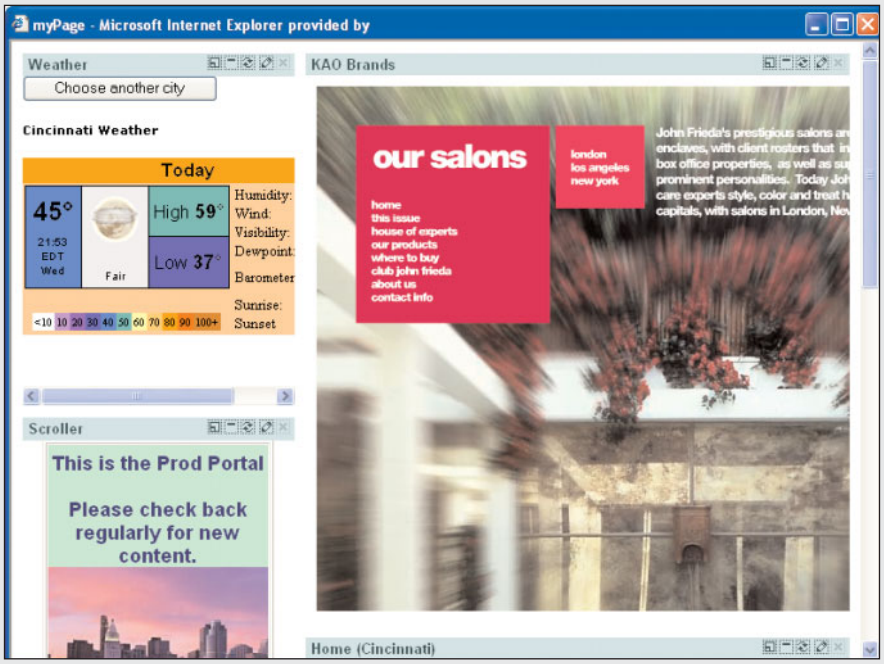


The final iView that references the URL parameters can be viewed using the preview option, and then added to a page for inclusion in the portal. The screenshot below shows an iView that uses parameters to display a personalized version of the Yahoo! weather program.



4. Test the iView by including it on a portal page, attaching the portal page to a user role, and logging on to the portal via a user ID associated with that role.

Figure 17 The Finished Portal Page



> ✓ ***Note!***
>
> *When intranet content is made available in an Internet-accessible portal, ensure that the content is placed on an Internet-accessible Web server! It sounds obvious, but I've found dead-end links on SAP's support portal (http://service.sap.com) — the URLs suggest that these documents are on an intranet-only document share. Inversely, make sure that links to internal documents and applications aren't included in external user pages, so external users don't come across similar "dead" links.*

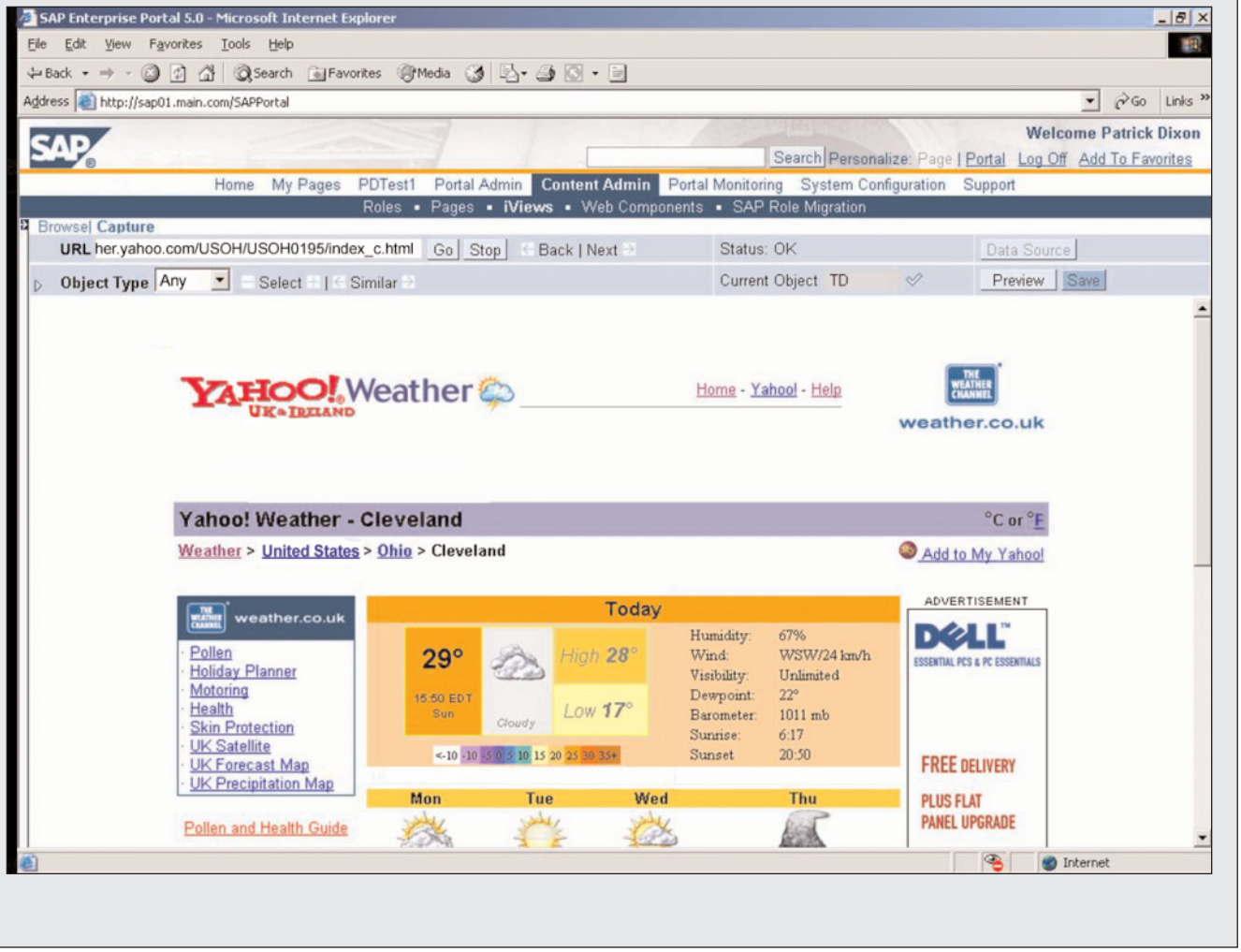
Figure 17 shows a finished portal page containing the example iView defined in Figures 15 and 16.

That's all there is to it! As you can see, this is a very quick and low-cost way to bring all of the Web content and applications a user works with into one place. If you've Web-enabled any of your third-party systems, just integrate a link to the login page (e.g., http://mydocumentmanagement/login) within the iView. Alternatively, you may be able to configure the backend system for single sign-on, which would log in the user automatically.

Use the SAP EP 5.0 iView Catcher to Extract Portions of Web Pages

Often Web pages do not look ideal within an iView, as the iView is much smaller than the Web page being

Figure 18 The iView Catcher Interface (SAP EP 5.0)

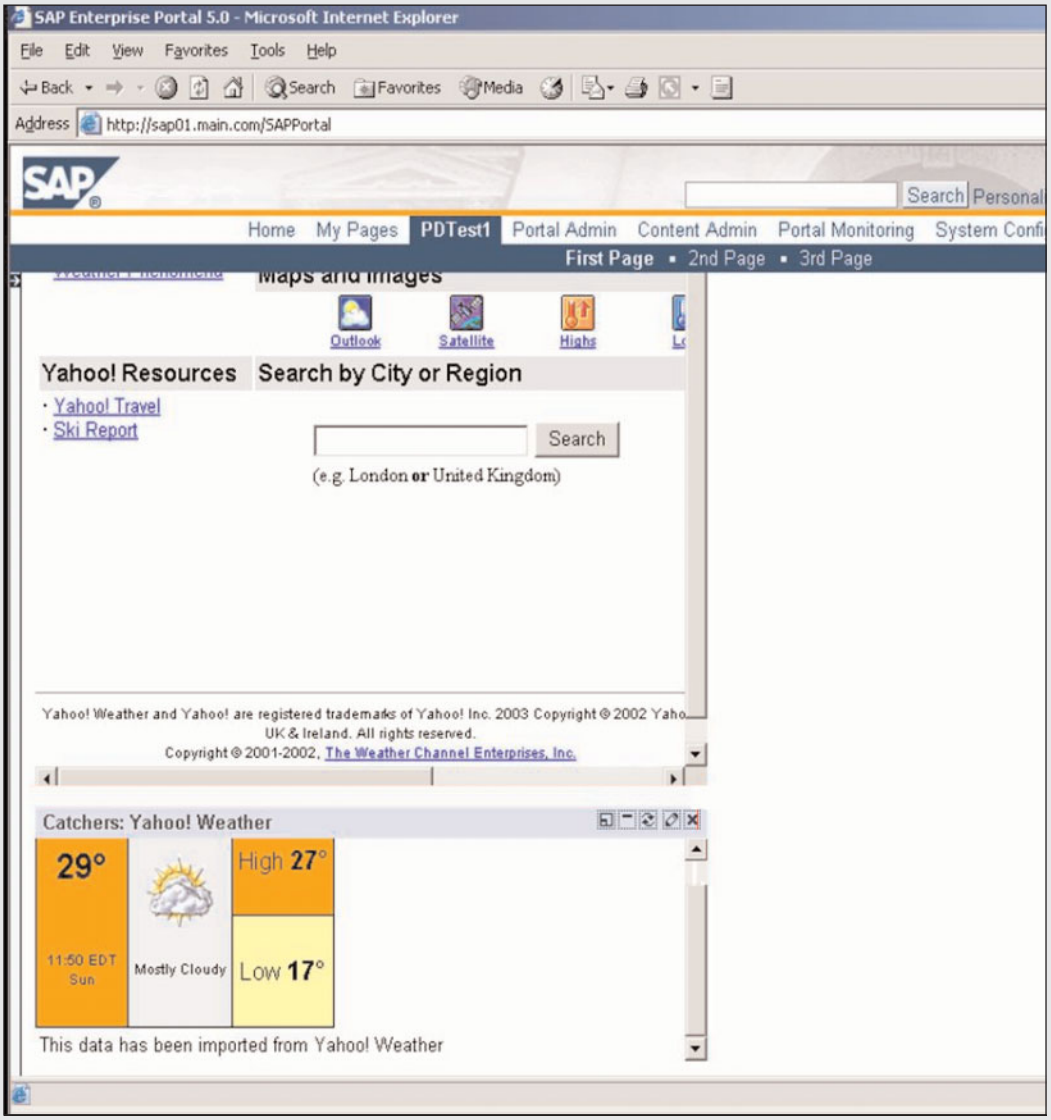


displayed. Furthermore, often only part of the information is needed on the Web page. The iView Catcher in SAP EP 5.0 (option ❸ in Figure 13) offers a way to do this (remember that the iView Catcher was removed from SAP EP 6.0 with the shift to SAP Web AS, as the iView Catcher relies on Microsoft components).

The iView Catcher works by parsing (or changing) the HTML on the Web page to XML, cutting out the parts of the page that are required, and then converting the XML back to HTML. Parsing assumes that the page layout does not change with time. If the page layout does change, the iView will yield strange results.

To access the iView Catcher, select *Content Admin → iViews* from within the portal administration tool, and then choose iView Catcher from the dropdown (refer back to Figure 3). You then specify the URL of the source Web page to capture, drag the mouse over the part of the page that you want to capture, and click on the Save button (see **Figure 18**). This creates an iView that you can place on user portal

Figure 19 An iView Developed with the iView Catcher



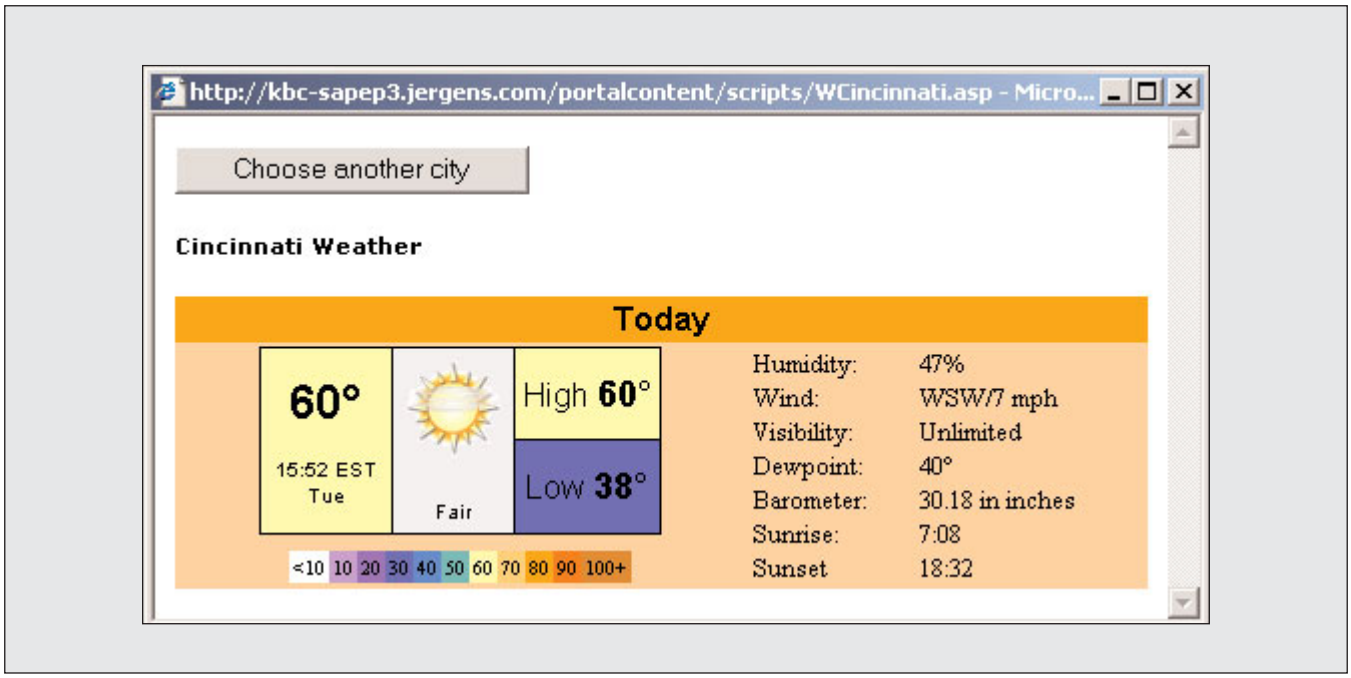
pages. **Figure 19** shows an example iView generated by the iView Catcher.

Develop Your Own XML Parser to Extract Portions of Web Pages

If you're running SAP EP 6.0, or aren't satisfied with the iView Catcher, consider developing your own parser — it's easy. The iView Catcher has two disadvantages: (1) it inserts a disclaimer at the bottom of the iView that can be irritating to internal users, who don't require the same legal treatment as external users; and (2) the catching is quite crude — you often either get too much or too little of the page.

Both disadvantages can be overcome by creating

Figure 20 An iView Developed Using the Example ASP-Based Parser



your own parser. You can also create a custom parser if you're running SAP EP 6.0 and have a Microsoft IIS Web server on which to host it. It is relatively simple and uses the Microsoft XML parser, which the iView Catcher uses as well.

Since I'm most familiar with ASP, I'll use ASP to create an example parsing program. The key elements of the ASP parser program are:

- The parser object:

```
Set objHTTP =
CreateObject("MSXML2.ServerXMLHTTP")
```

- The parsed URL:

```
sURL = "http://uk.weather.yahoo.com/
USOH/USOH0195/index_f.html"
```

- The start/end point of the HTML code we want to keep:

```
iPos =
InStr(1,UCASE(sBody),"TODAY",1)
```

The result is that we have the data we want — e.g., the humidity value in **Figure 20** — without the disclaimer. Note that parsing Internet content in this way can be legally dubious without the explicit permission of the content provider (it's called "inlining" in the legal world). Seek guidance from your legal department.

Conclusion

This article has completed our tour of integration options and strategies for SAP Enterprise Portal 5.0 and 6.0. The first installment, published in the July/August 2004 issue of this publication, explored your options for integrating SAP-based content. This second installment explored specialized techniques for non-SAP and Web-based content, including email

systems, SQL databases, and Web pages. As a next step, I encourage you to try out each of these integration options on your own, and use the knowledge you've gained to plan and prioritize your portal content offerings. In my experience, user perception of the portal's value is tied almost entirely to its content (i.e., vs. look and feel or even performance), and is the single most important determinant of portal adoption.

Patrick Dixon is a manager with Deloitte specializing in SAP Enterprise Portal and Master Data Management (MDM). He has over five years of experience Web-enabling and integrating SAP systems with SAP Enterprise Portal, SAP Internet Transaction Server (ITS), SAP Exchange Infrastructure (XI), and Plumtree Software solutions. Patrick was a key speaker on portal implementation and content integration at the SAP BW and Portals 2004 and SAP Portals 2003 events. Prior to moving to the US in 1996, he was a freelance consultant in Great Britain. Patrick graduated from Bath University in 1983 with a master's degree in computer simulation, and subsequently obtained an MBA from City University, London. You can reach him at padixon@deloitte.com.

