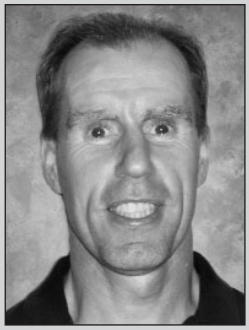
Integrating SAP Transactions, Reports, and Data into Your SAP Enterprise Portal — A Guided Tour of Your Options, Which to Use, and When

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 76)

What is the main function of a portal? Presentation? Deployment flexibility? Certainly these are important, but the real value of a portal is its ability to bring together data, content, and applications from SAP, non-SAP, and Web-based systems into a single, organized, seamless interface whose underlying complexity is transparent to end users. With SAP Enterprise Portal (SAP EP), this value is arguably most apparent to end users, who receive and interact with content from any number of systems through one or more iViews[1] on a portal page, or through an external service[2] displayed in a separate browser window. Users also benefit from an organized, consistent navigation framework built around a global header and navigation panel, and powerful personalization options like the ability to adjust page layouts and maintain lists of single sign-on passwords.[3]

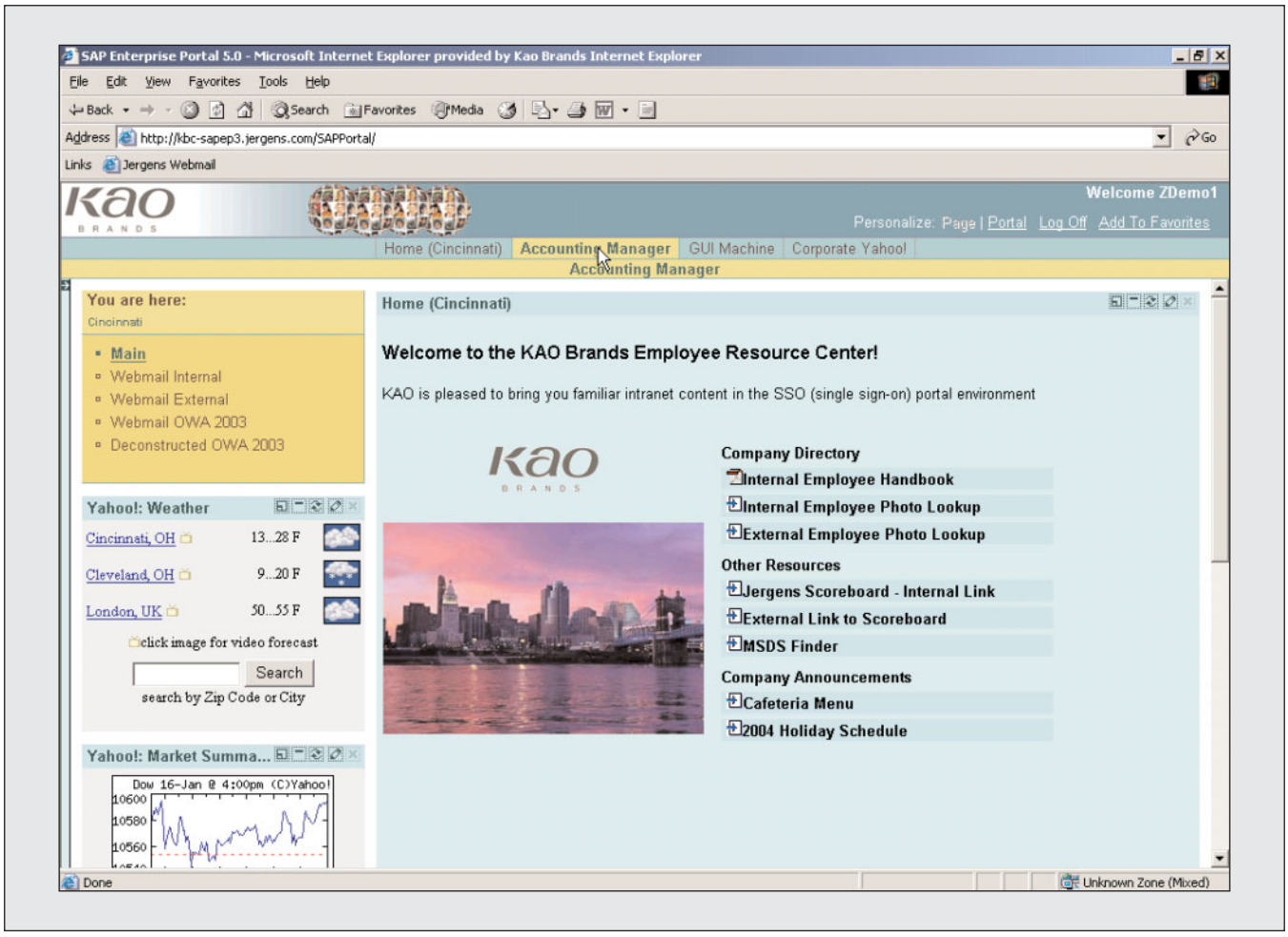
While simplified navigation and personalization features are powerful advantages for end users, for developers and administrators much of SAP EP's power lies behind the scenes — in its suite of wizards, tools, and predeveloped iViews for integrating content from SAP, non-SAP, and intranet/Internet sources. This article will take you on a guided tour of the options available to you in SAP EP 5.0 (support package 5

[1] An iView is a small pane on a portal page through which users receive data or content, or interact with an application. For more on iViews, see the article "Build Custom Java iViews for SAP Data Using Eclipse: A Guide for Developers and Implementation Teams" published in the March/April 2003 issue of *SAP Professional Journal*.

[2] The external services feature is available with SAP EP 5.0 only. See the sidebar on pages 45-46 for details.

[3] Single sign-on is an optional feature in which the portal automatically signs on to a backend system when its content is requested. The user is not prompted to log in.

Figure 1 A Portal Page Built with the SAP EP Toolset



and higher) and 6.0[4] for integrating SAP transactions, reports, and data into a portal, and the unique challenges associated with each. I will provide you with an overview of the options, along with the pros and cons of each, and explain how they can be used to integrate your existing SAP content and applications into a portal like the one shown in **Figure 1**. Along the way, I'll point out which options are available only in SAP EP 6.0, and which 5.0 options have been discontinued or reimplemented in 6.0 (see the sidebar on the next page for a summary of their key differences).

Before we discuss specific SAP content integration techniques, however, it's important that you understand the "big picture" of what your portal can do. This will help you more easily digest and contrast the large number of options I'll discuss later in the article, so you can make the right decision about which options will work best for your own content integration project.

> ✓ ***Note!***
>
> *Some of the examples and content in this article were featured at SAP BW and Portals 2004 in Orlando, Florida. For more information on this event, visit* ***http://sapbwportals2005.com****.*

[4] SAP EP 5.0 SP5 and SAP EP 6.0 are the most widely used SAP portal platforms. Note that for clarity, I will focus on integrating content with the base SAP EP server configuration. I will not discuss Knowledge Management and Unification — two optional portal add-ons — in detail. For more on these, visit the SAP online help at **http://help.sap.com/portals**.

SAP Enterprise Portal 5.0 vs. 6.0

From a content integration perspective, SAP Enterprise Portal 5.0 (support package 5 and higher) and 6.0 are conceptually similar, but their underlying architectures and user interfaces differ a bit. SAP EP 6.0 adds a few new integration options and removes a few staples of the 5.0 world. Here are the changes you need to know about:

☑ **The underlying platform is SAP Web Application Server (SAP Web AS) in SAP EP 6.0.** SAP EP 5.0 relies on two servers to deliver content: Microsoft Internet Information Server (IIS) for the Web server, and the SAP Java 2 Platform, Enterprise Edition (J2EE) server for the portal runtime (see the diagram on the next page). SAP EP 6.0, however, runs solely on SAP Web AS,* which includes both a Web server and the Java (J2EE) runtime. For you, this means more reliable operation, a single vendor for support and updates, and the option to run your portal on a non-Windows system — SAP Web AS (and thus SAP EP) runs on both Unix and Windows. Keep in mind that it also means that the 6.0 portal Web server can no longer run Active Server Pages (ASPs), .NET, or SAP EP 5.0 tools like the iView Catcher, which rely on Microsoft components, on its own.

> ✓ ***Tip***
>
> *To run ASP/.NET applications on a Windows-based system, you'll need to run Microsoft IIS either in parallel to the portal Web server (on a different port), or on a different box.*

☑ **The "iView Editor" wizards are replaced by "templates" in SAP EP 6.0.** SAP EP 5.0 includes "iView Editor" wizards for defining four types of iViews: .NET Unifier, .NET Custom, Java, and iView Catcher. The Java iView wizard offers an additional eight iView subtypes to choose from. SAP EP 6.0 uses only one iView type — Java — and instead of the iView Editor introduces the concept of creating iViews from "templates." The iViews created from a particular template are tied to that template by a delta link, and you can specify that an iView inherit all or part of a template's attributes.

☑ **Some SAP EP 5.0 business packages will not run on 6.0.** Some business packages that run on SAP EP 5.0 will also run on 6.0, even though they are not "officially" compatible with 6.0.** Others, particularly those that need .NET support, will not work at all on 6.0. Among those that won't work is the Communication business package, which contains iViews for accessing mail from Microsoft Exchange.*** Since you cannot uninstall business packages, consider trying out any 5.0 packages

---

* SAP Web AS underlies all SAP NetWeaver products like mySAP ERP, BW, etc. All that's required to run a portal is SAP Web AS with the Java runtime installed.

** That is, 6.0 is not included in the Portal Release field on the package's detail page at **www.iViewStudio.com**. Note that as of July 2004, the business packages are also listed at the SAP Service Marketplace (**http://service.sap.com/ep-content**), which has been redeveloped as an SAP EP 6.0 portal.

*** The good news is that SAP EP 6.0 has a built-in connector to Microsoft Exchange (a connector to Lotus Notes is currently under development). To configure an email transport in SAP EP 6.0, from the top-level navigation bar choose *System Administration → System Configuration → Knowledge Management → Configuration → Collaboration → Transports → Mail Transport*, and click on *New* to create a new transport instance. Next, to configure the email service, choose *System Administration → System Configuration → Knowledge Management → Configuration → Content Management → Global Services → Mailing Services* from the configuration iView, and edit the default instance.

(continued on next page)

(continued from previous page)

you are uncertain about on a sandbox system running 6.0 before installing them on your development system. See SAP Note 642775 for details on determining which SAP EP version a business package applies to.

☑ **SAP EP 6.0 includes a visual tool for defining available systems.** A visual tool included with SAP EP 6.0 helps you maintain the system definitions file (portalapp.xml), which determines the backend systems you can choose for the portal. In SAP EP 5.0, the system definitions are stored in two files — systems.xml and JCODestinations.xml — which you need to download from the portal content directory on the portal server, edit manually, and then upload back to the portal content directory (for more on this, see the download available at **www.SAPpro.com**).

☑ **In SAP EP 6.0, the portal content directory resides in a database.** In SAP EP 5.0, the portal content directory is file-based and vulnerable to any changes made to the J2EE directory during upgrades. In SAP EP 6.0, the portal content directory is located in the portal database to reduce this vulnerability, simplify administration, and improve scalability.

☑ **The external services feature is discontinued in SAP EP 6.0.** In SAP EP 5.0, you can direct content to an "external service" defined in a separate browser window instead of to a traditional portal page. While this is a great option for applications that need extra real estate, the separate browser window lacks the standard portal header and navigation controls, which can strand or confuse users. Presumably for this reason, SAP discontinued the feature in SAP EP 6.0. For consistency, I advise both my SAP EP 5.0 and 6.0 clients to avoid using external services, and instead define an additional portal page with a single iView. This approach also allows you the flexibility to add further iViews to the page if needed; a single external service, in contrast, takes up an entire browser window.

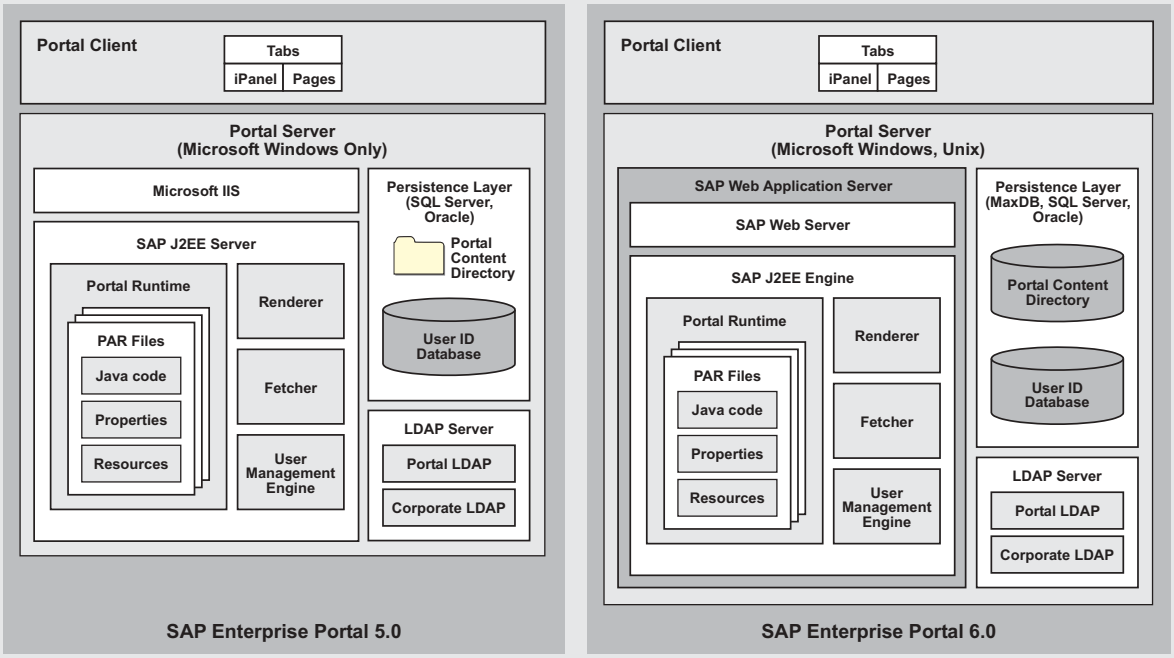
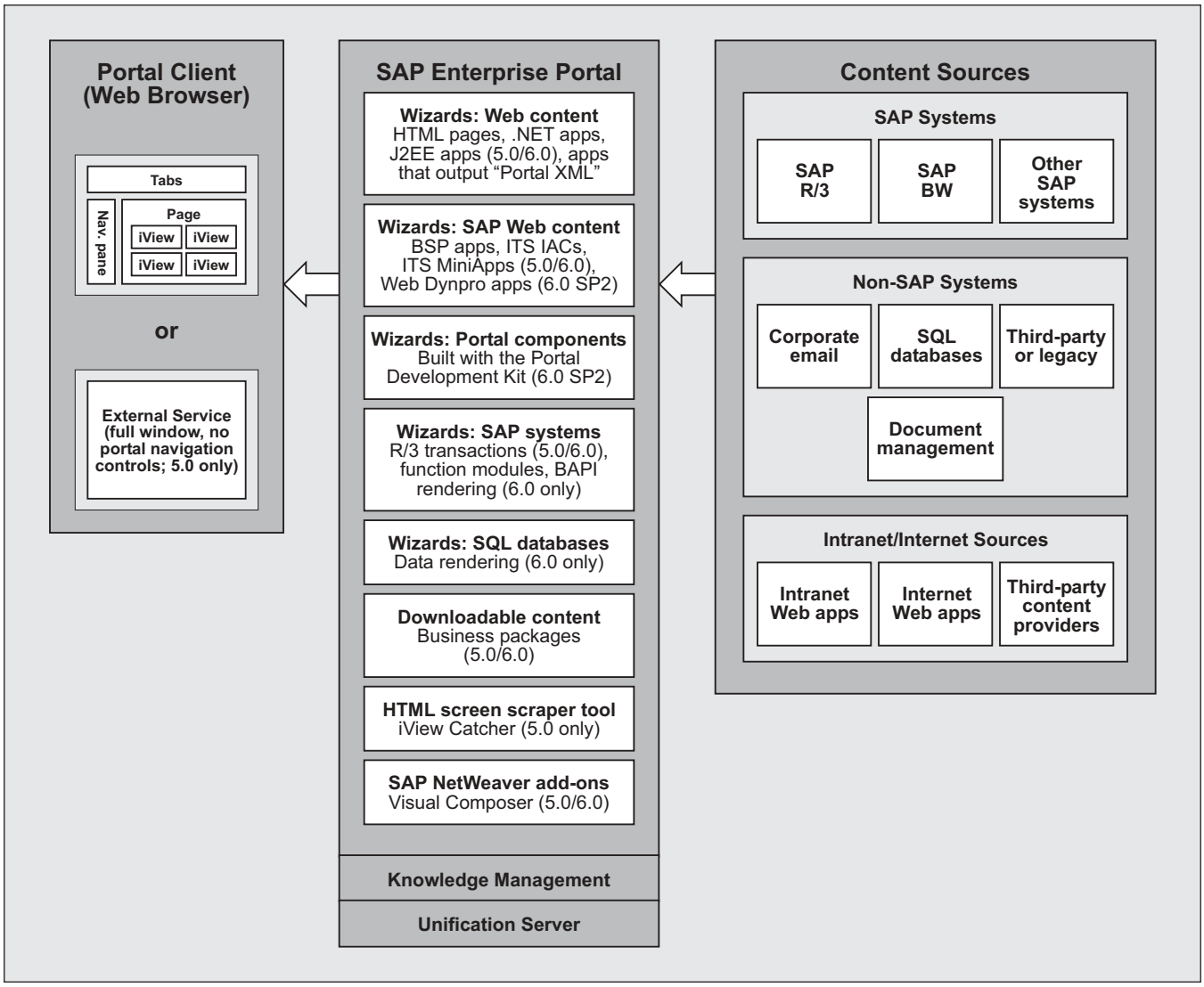


Figure 2 Integrating SAP, Non-SAP, and Web-Based Content with the SAP EP Toolset

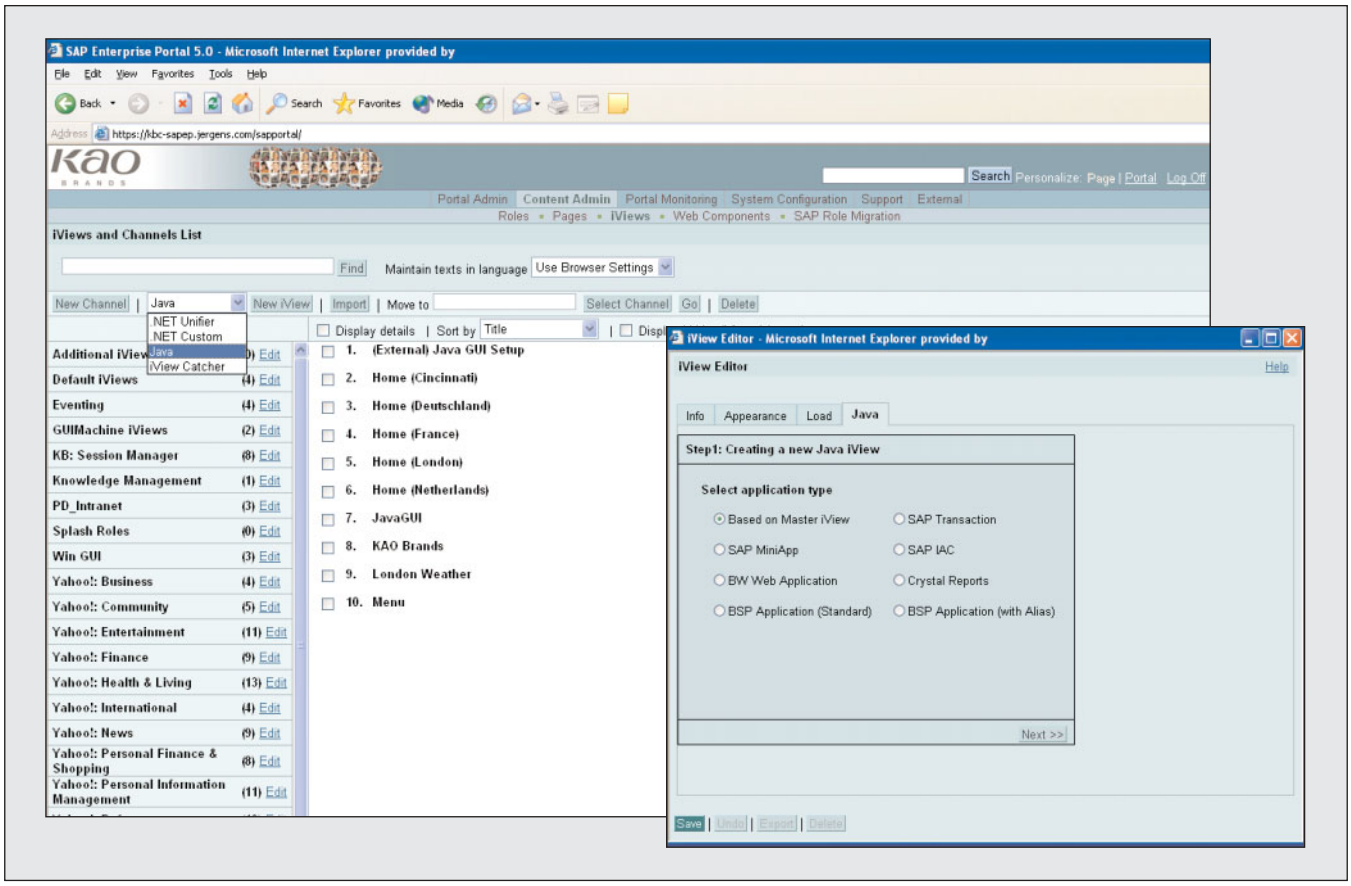


An Overview of the Portal Content Integration Toolset

The SAP EP 5.0 SP5 and 6.0 toolset is organized into wizards, tools, and downloadable content that enable you to integrate SAP, non-SAP, and intranet/Internet-based content into a single portal. **Figure 2** provides a graphical overview of integrating content using the SAP EP toolset (note that the Knowledge Management and Unification portal add-ons shown in the diagram are beyond the scope of this article, as noted in footnote 4). In the following sections, we'll take a closer look at the various components of the toolset:

- Generic wizards for Web-based applications and content, including support for applications that output "Portal XML"
- Specialized wizards for SAP Web-based content
- Wizards for Java "portal components"
- Specialized wizards for SAP systems and SQL databases
- Downloadable business packages

Figure 3 Creating an iView with SAP Enterprise Portal 5.0



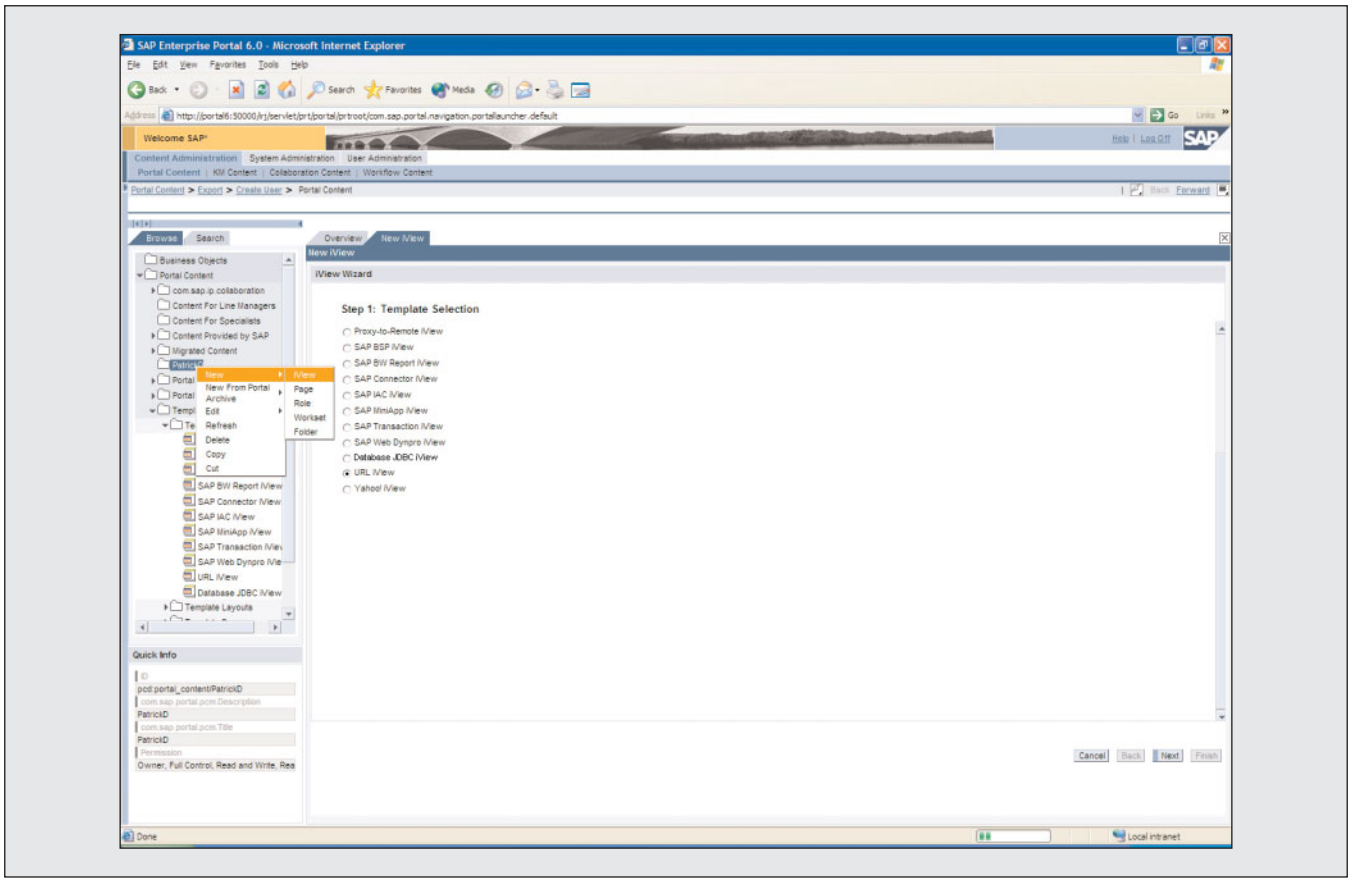
- A Web page "screen scraper" tool (available only with SAP EP 5.0)
- New SAP NetWeaver add-ons

Generic Wizards for Web-Based Applications and Content, Including Support for "Portal XML"

Any application or content that can be accessed via a URL can be integrated into your portal. A generic URL iView wizard included in both SAP EP 5.0 and 6.0 demonstrates this — you just specify a URL, and the portal fetches it at runtime. In SAP EP 5.0, generic URL iViews are called .NET Custom iViews, and are created in the iView content administration screen (shown in **Figure 3**) by choosing the *.NET Custom* option from the dropdown list at the upper left. In SAP EP 6.0, generic URL iViews are simply called URL iViews, and are created by choosing the *URL iView* template from the list of iView templates (see **Figure 4**).

As you can see, SAP EP 5.0 and 6.0 use slightly different approaches and terminology when defining new iViews. In SAP EP 5.0, you start by selecting *Content Admin* → *iViews* at the top of the iView content administration screen, and then select an item from the dropdown menu (as shown at the upper left in Figure 3). Selecting the iView Catcher item launches a Web "screen scraper" tool that enables you to capture specific elements and data from a Web page (I'll discuss this tool later in the article). Selecting the .NET Unifier, .NET Custom, or Java item launches that item's iView Editor wizard, which then guides you through the process of defining the iView. The .NET Unifier option is for creating Drag&Relate iViews by linking the unification data source to the iView (see the sidebar on pages 50-51

Figure 4 Creating an iView with SAP Enterprise Portal 6.0



for more on this); the .NET Custom option, as mentioned previously, is for creating iViews based on URLs.[5] The Java option is for creating Java-based components; the first screen of the Java iView wizard, also shown in Figure 3, then asks you to choose from eight different iView subtypes.

SAP EP 6.0 does away with the distinction between .NET and Java iViews[6] and introduces the concept of iView "templates," which allow administrators to define a set of baseline attributes, and then selectively inherit or override them at the iView level.[7]

The iView content administration screen was also redesigned to use a list-based interface (see Figure 4). To define a new iView in SAP EP 6.0, you right-click on the portal content directory[8] folder you want to add it to and choose *New → iView*, as shown in the left pane of Figure 4. Alternatively, you can base the iView directly on a portal archive (PAR) file (in 5.0, you can do this indirectly by selecting the *Based on Master iView* subtype for a Java iView).

There is one extra option to note here: when writing Web-based applications, you can have SAP EP 5.0 and 6.0 render the HTML interface for you by outputting "Portal XML" instead of HTML. Portal XML, which was previously called "BusDoc XML," consists of a set of predefined XML tags that tell the portal to generate headers, multi-cell rows, etc. The upside to

[5] Note that the division between the .NET Unifier, .NET Custom, and Java iViews in SAP EP 5.0 parallels the 5.0 portal's architecture, which includes a unification server, a Microsoft IIS capable of running .NET applications, and the J2EE runtime on which the portal server resides.

[6] Mainly because native .NET support was "discontinued" with SAP EP 6.0's move to SAP Web AS, where all iViews are Java iViews.

[7] Inherited attributes are instantly updated if any values change in the template.

[8] The portal content directory stores iView content, page and role definitions, relationship mappings, and user personalization settings.

Enabling Drag&Relate with the Unification Server

Unification is a sizeable topic that could easily justify its own article, so this article does not cover ".NET Unifier" (SAP EP 5.0) and "Unifier Project" (SAP EP 6.0) iViews. For your reference, however, here are the key points you need to know:

☑ The unification server is an optional component included with the SAP EP 5.0 and 6.0 CDs that is installed and run separately. It can be integrated with SAP EP 5.0 and 6.0 and has its own release cycle — as of this writing, the current release of the unification server is 5.0.

☑ Unification's main feature is the "Drag&Relate" technology, which enables you to define hotspots onto which users can drag and drop values on their pages. In this way, users can quickly display a sales order, for example, by dragging and dropping a sales order number from an order status list to a link that says "Display order." Upon dropping the number onto the link, the browser notifies the unification server of the number and the link it was dropped onto via HTTP, and the unification server takes whatever action you have configured it to take. In this case, for example, you might have configured the server to instruct the browser to navigate to a portal page containing an SAP Transaction iView* defined to display VA03 (Display Sales Order), and to pass along the order number in the URL.

Setting up Drag&Relate can be challenging and tedious. First, you need to drag-and-drop-enable all objects that you want to be "draggable" in the source application. There are two ways to do this — build the relationships yourself based on the data stored in a SQL database, or use one of the SAP Unifiers, which constructs the relationships on the native system (SAP R/3, SAP BW, etc.) and then uses the unification server to enable the Drag&Relate mechanism:

- **Building the relationship yourself:** To do this, you need to define a "unifier project," along with the fields that will be the source and object of the draggable action. This determines the targets onto which draggable objects can be dropped. Lastly, you need to define the positions of these

* A type of iView based on the SAP Transaction Java iView subtype (in SAP EP 5.0) or on the SAP Transaction iView template (in SAP EP 6.0).

this approach is that all of your Portal XML applications share the same look — they are all rendered by the portal. Also, when a Web page is called, the portal automatically renders elements defined using Portal XML, such as table lists, so you do not have to code control elements like table scrolling. The downside of Portal XML is that Internet browsers cannot render Portal XML, so you will only be able to access the application through the portal. There's also a limited set of tags, so your design options are restricted. In SAP EP 5.0, Portal XML iViews are called .NET Rendered iViews and are defined using the .NET Custom iView Editor wizard — there's a special checkbox within the wizard to mark the iView as "rendered." In SAP EP 6.0, they're simply called Rendered iViews, and are defined using the URL iView template, or a template based on an imported Java PAR file.

objects in the iPanel.** The unifier project is linked to the portal using a "data object," which is then incorporated into the unification iViews.

- **Using one of the SAP Unifiers:** In addition to Drag&Relate, the unification server includes optional add-in modules called "Unifiers." The 5.0 unification server comes with two Unifiers: the SQL Unifier and the R/3 Unifier, although only the SQL Unifier is free of charge. Unifiers for other enterprise systems like Oracle are available on request from SAP. Using a Unifier involves a three-part configuration process: (1) install the Unifier on the native system; (2) create the unifier project on the unification server; and (3) set up the unification iViews on the portal.

☑ In SAP EP 5.0, unifier projects are integrated into the portal from within the unification server. Unifier projects must be defined in the portal as "data sources," and SAP unifier projects (SAP R/3 and SAP BW) must also be defined as systems in the portal systems.xml file. A "data source" is really just an object that tells the iView where the unification server is keeping the Drag&Relate details. For detailed information, go to the SAP EP 5.0 documentation at **http://help.sap.com/portals** and navigate to *SAP Enterprise Portal Documentation → Administration Guide → Integration of Applications and Data Sources → Data Sources and Mapping*.

In SAP EP 6.0, all unifier projects are defined in the portal as "systems," to integrate them with the portal and make their components available as iViews. For detailed information, go to the SAP EP 6.0 documentation at **http://help.sap.com/portals** and navigate to *SAP Enterprise Portal Documentation → Administration Guide → Unification → Creating a Unifier Project*.

The bottom line with Drag&Relate is that it involves a significant learning curve on top of the portal administration and configuration knowledge that is required. It also involves additional hardware, a database, and connections for the unification server. While eventing can provide better navigation without extra hardware additions, it involves custom coding. Unification, on the other hand, does not, and Drag&Relate iViews can be developed very quickly once the unification server is configured and the data to be made draggable is added to the database.

** The iPanel is a tree for Drag&Relate navigation that appears on the left side of the portal page.

✓ ***Note!***

Since this article discusses both SAP EP 5.0 and 6.0, I will use the terms "wizard" and "template" interchangeably. In SAP EP 5.0, it is accurate to say that an iView is created via a particular wizard (e.g., the Java iView wizard or the .NET Custom iView wizard). In SAP EP 6.0, it is more accurate to say that an iView is created based on a particular template, even though the process is still wizard-based. In both versions, it is also accurate to say that an iView is of a particular type (e.g., a Java BSP Application iView or a .NET Custom iView).

Specialized Wizards for SAP Web-Based Content

In addition to a generic URL wizard, SAP EP also includes wizards with specialized options for Web-based applications built with SAP technologies, such as BW Web applications or reports,[9] Business Server Pages (BSP)[10] applications running locally or remotely on SAP Web AS, and Internet Application Components (IACs)[11] running on an SAP Internet Transaction Server (ITS). In SAP EP 5.0, you'll encounter these application types when defining a Java iView (see the list of subtypes in the screenshot in Figure 3). In SAP EP 6.0, SAP has defined a standard iView template for each type, so you'll find them in the list of iView templates that appears when defining a new iView (see Figure 4).

Wizards for Java "Portal Components"

"Portal components" are mini-Java applications built with the Portal Development Kit (PDK)[12] that run within a container called the "portal runtime." SAP introduced the portal components development model to make building custom portal applications easier.[13] A specialized wizard lets you define iViews based on the portal archive (PAR) files generated and installed by the PDK. In SAP EP 5.0, portal archives are automatically added to a list of "master iViews" (see **Figure 5**). To create a new portal component iView, you select the *Based on Master iView* option in the Java iView Editor dialog (see Figure 3). SAP EP 6.0 does away with the confusing master iView concept — you now base iViews directly on a portal archive by right-clicking on a folder in the portal content directory and selecting *New From Portal Archive → iView* (just below the *New* menu item shown in Figure 4).

> ✓ ***Note!***
>
> *In addition to portal components, SAP EP 5.0 and 6.0 can host native Java applications (JSPs, EJBs, etc.) on the SAP Web AS J2EE runtime, as well as .NET applications if Microsoft IIS is running on your portal server.[14] If you are running at least SAP EP 6.0 with support package 2, your server can also host custom Web Dynpro applications[15] developed with the PDK. From a content integration perspective, each of these applications is treated like a remote, Web-based application, so you integrate them using a generic URL iView.*

> ✓ ***Tip***
>
> *The subtypes listed by the SAP EP 5.0 Java iView Editor wizard (see Figure 3) are actually portal components in disguise — each has an entry in SAP EP 5.0's list of master iViews. SAP made instantiating these popular master iViews easier by adding explicit options for them on the main wizard screen.*

---

[9] BW Web applications are Web-based queries designed with the PC-based Web Application Designer tool (program *wdbpwpub.exe*) that comes with SAP BW 3.0 and higher.

[10] Business Server Pages (BSPs) are dynamic Web pages that are written and run on SAP Web AS. They are analogous to JavaServer Pages (JSPs) and Active Server Pages (ASPs). For a detailed introduction to BSPs, see the article "A Developer's Guide to Creating Powerful and Flexible Web Applications with the New Web Application Builder" in the January/February 2002 issue of *SAP Professional Journal*.

[11] Also known as Easy Web Transactions (EWTs).

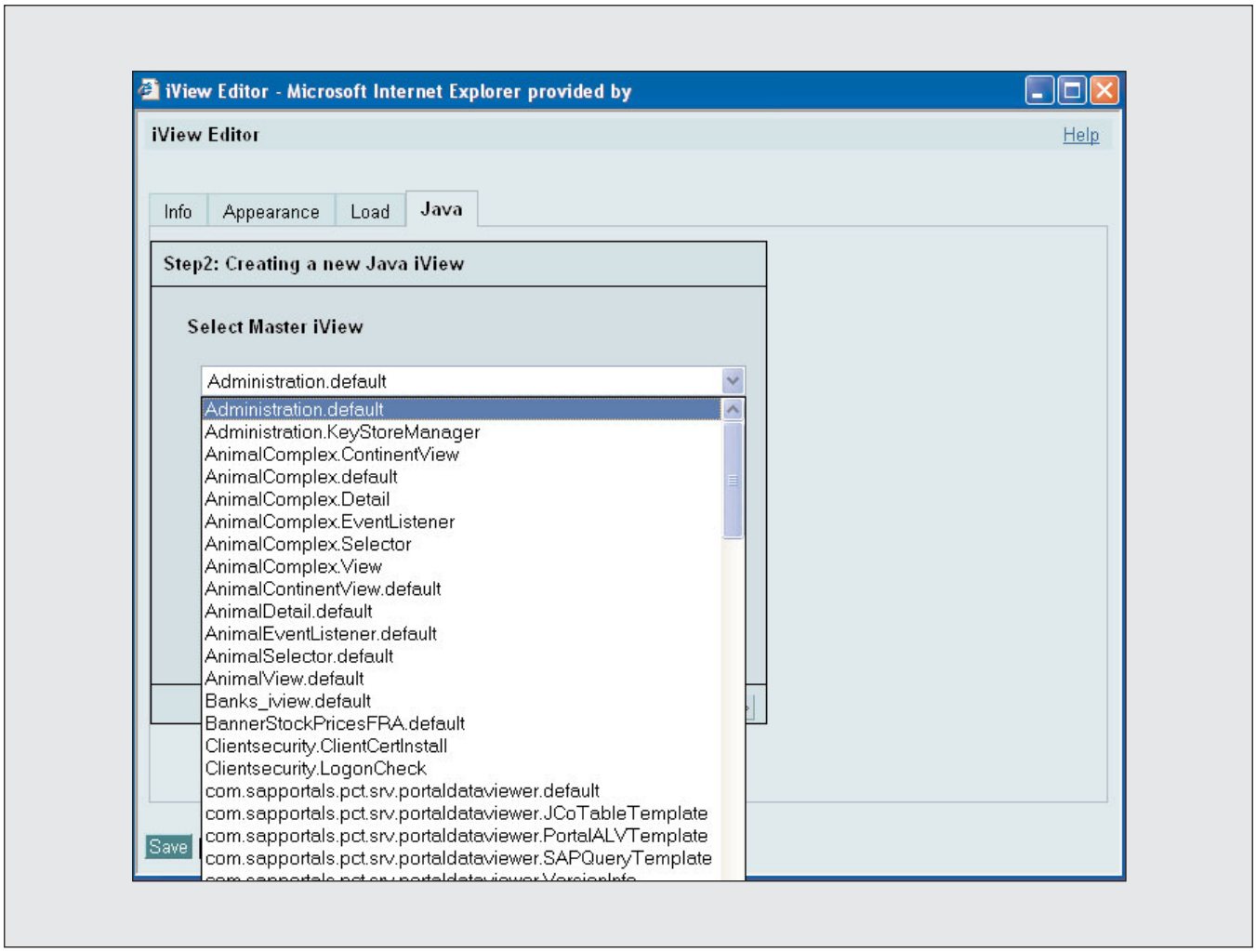
[12] The Portal Development Kit (PDK) is available from **www.iViewStudio.com**. For more on the PDK, see the article "PDK Installation and Customization for SSO Access to SAP Systems: Essential Lessons for Developers and Implementation Teams" in the November/December 2002 issue of *SAP Professional Journal*.

[13] Portal components are collections of Java components (e.g., JSPs) and associated images, files, etc. Coding is simplified through the use of predelivered HTML Business tags that automatically generate the HTML code for complex user interface objects like tables, graphs, and trees.

[14] SAP EP 5.0 uses Microsoft IIS as its Web server, so it natively supports .NET. SAP EP 6.0 uses a proprietary Web server built into SAP Web AS and can run on either Windows or Unix — you can only host .NET applications on your SAP EP 6.0 portal server if you are running on Windows and are running a Microsoft IIS Web server in parallel.

[15] Web Dynpro is a new visual programming model for building Web-based business applications, available with SAP Web AS 6.40.

Figure 5 *SAP EP 5.0 Master iViews*



Specialized Wizards for SAP Systems and SQL Databases

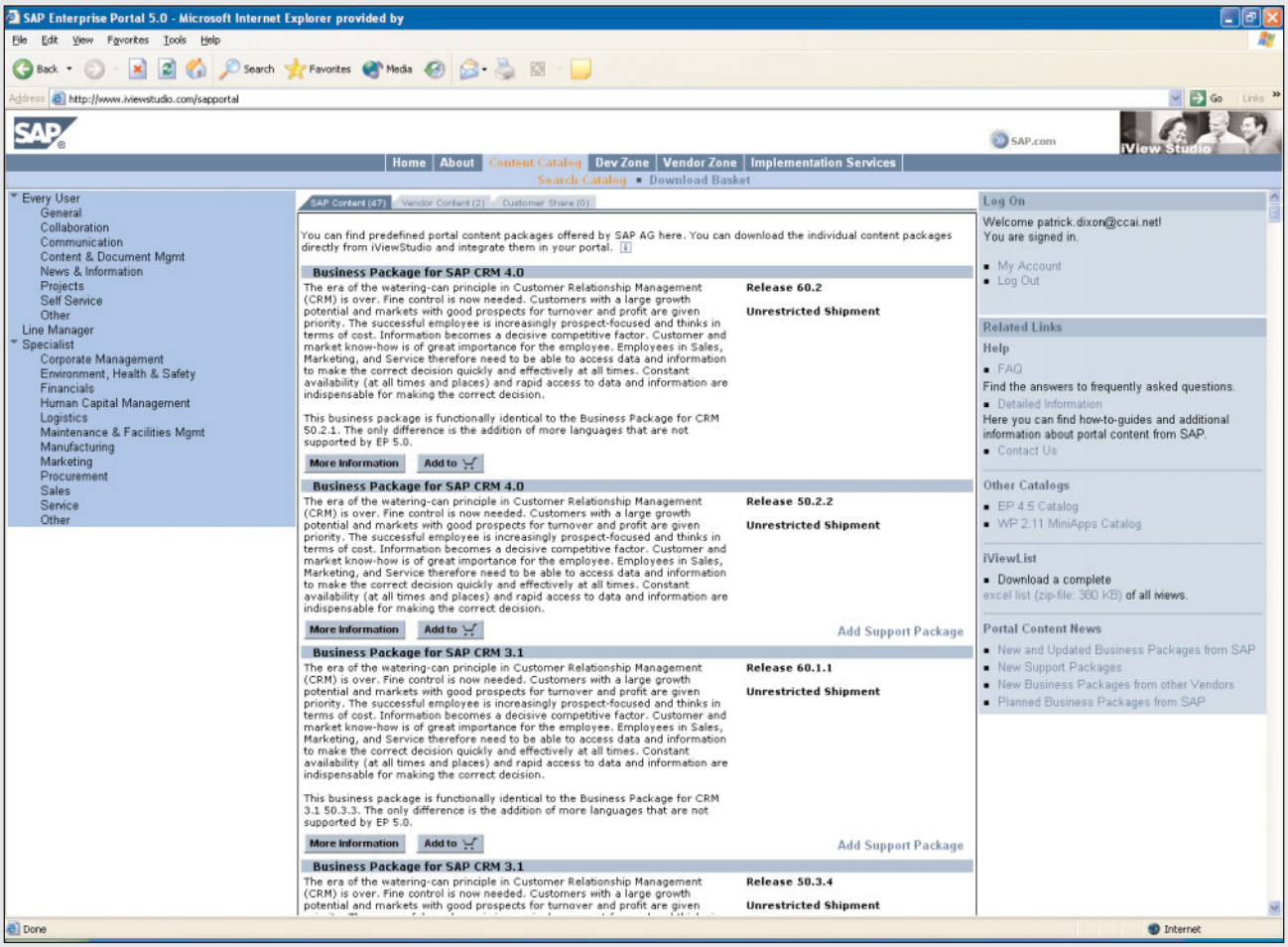
SAP EP 5.0 and 6.0 each provide a powerful SAP Transaction iView wizard (see Figures 3 and 4) you can use to instantly deploy SAP transactions to portal users — simply enter an SAP transaction code and specify which SAPGUI the iView should use,[16] and the portal launches the SAPGUI within the user's browser. In addition, SAP EP 6.0 includes two new iView types, SAP Connector and Database JDBC, that *automatically* display data from SAP R/3 function modules and SQL databases, respectively! I'll discuss how to use each of these in more detail later.

[16] SAPGUI for Windows (WinGUI), SAPGUI for Java (JavaGUI), or SAPGUI for HTML (WebGUI). I'll explain each of these and when to choose which later in the article.

Downloadable Business Packages

Business packages are collections of iViews, pages, and roles developed by SAP or third-party organizations that are available for download from **www.iViewStudio.com**. This Web site provides

Figure 6 Business Packages Available from iViewStudio's Content Catalog



a searchable catalog (see **Figure 6**) of dozens of out-of-the-box applications, and usually contains enough documentation and sample screenshots to help you evaluate a package before downloading it. Installing business packages in your portal is easy — just download the package to a temporary directory on your portal server, import it from within the Web-based portal administration tool (*Portal Admin → Import*), and configure the package as described in the documentation.[17] Because business packages are predeveloped and supported by their creators, **www.iViewStudio.com** should always be one of your first stops in the search for new portal content.

A Web Page "Screen Scraper" Tool (Available Only with SAP EP 5.0)

SAP EP 5.0 includes a useful Web page "screen scraper" tool called the iView Catcher that lets you dynamically extract portions of content from Web pages or Web-based applications. The real value of

[17] Configuring the business packages is the time-consuming part! The documentation is accessible from, but not on, the package detail screen at **www.iViewStudio.com**.

this tool is that it enables you to leverage existing Web content *and* deliver only what the user needs to see (e.g., a dynamically generated cash forecast chart embedded in a complex Web page). The iView Catcher outputs iViews that can be added to your portal pages just like any other iView.

> ✓ ***Tip***
>
> *Be aware that the iView Catcher tool inserts a legal disclaimer at the bottom of each iView, since it could theoretically be screen scraping copyrighted content. This feature cannot be disabled.*

> ✓ ***Note!***
>
> *The iView Catcher is not available with SAP EP 6.0, since it relies on Microsoft components that are not included with 6.0. If you upgrade from SAP EP 5.0 to 6.0, keep in mind that any iView Catcher iViews you have defined will not run.*

New SAP NetWeaver Add-Ons

SAP is currently developing portal add-ons that offer new options for delivering SAP content through your portal. We'll review one of these tools (Visual Composer[18]) later in the article. Visual Composer renders a *customizable* HTML interface for BAPI data. You can keep abreast of which add-ons are available for your system, and download installation packages, at **http://sdn.sap.com**.

[18] Formerly known as the "GUI Machine," Visual Composer goes a step beyond the SAP Connector iView wizard by letting you completely customize the generated user interface using a WYSIWYG tool.

Using the SAP EP 5.0 and 6.0 Wizards, Templates, and Tools to Integrate Content

A key thing to understand is that the portal doesn't care which of these wizards, templates, or tools you use to generate an iView. The portal ultimately treats all iViews the same: it stores their definitions in the portal content directory, executes the Java (or other) component upon which the iView is based, and coordinates delivery of the resulting HTML/JavaScript to the browser.

Another thing to keep in mind is that there is often more than one way to integrate a particular type of content. For example, to build an iView that will access a BSP page, you can use the BSP Application iView wizard, or you can use the generic URL iView wizard (the URL iView template in 6.0; the .NET Custom iView Editor in 5.0) and hard-code the URL that will launch the application. The specialized wizards will almost always be easier to use, but occasionally you'll need full control over the parameters that are passed or how the URL is formed.

Now that you know the specific tools available to you that make content integration possible, let's look at the various techniques you can use to deliver content from your SAP systems through your portal. The sidebar on the next page outlines the system settings that must be in place before you can start using SAP EP to integrate content.

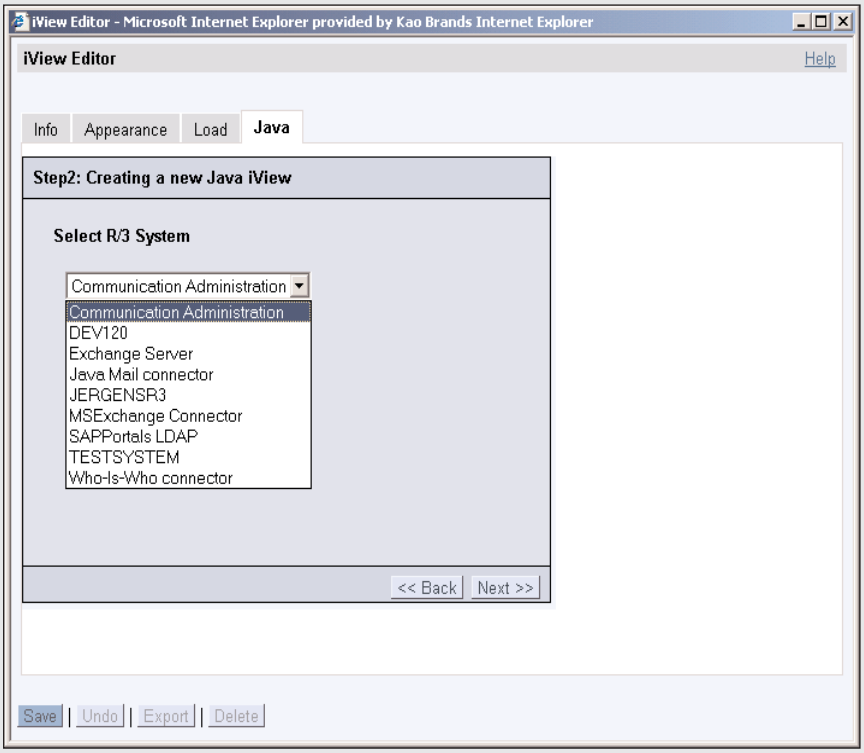
Integrating Your Existing SAP Content and Applications into a Portal

If your company is like most, a large portion of your portal project will involve integrating your existing SAP transactions, reports, and data. While SAP EP's integration tools make this easy from a technical standpoint, portal teams have a large number of options to digest and choose from, each with specific system and release requirements. Fortunately,

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.

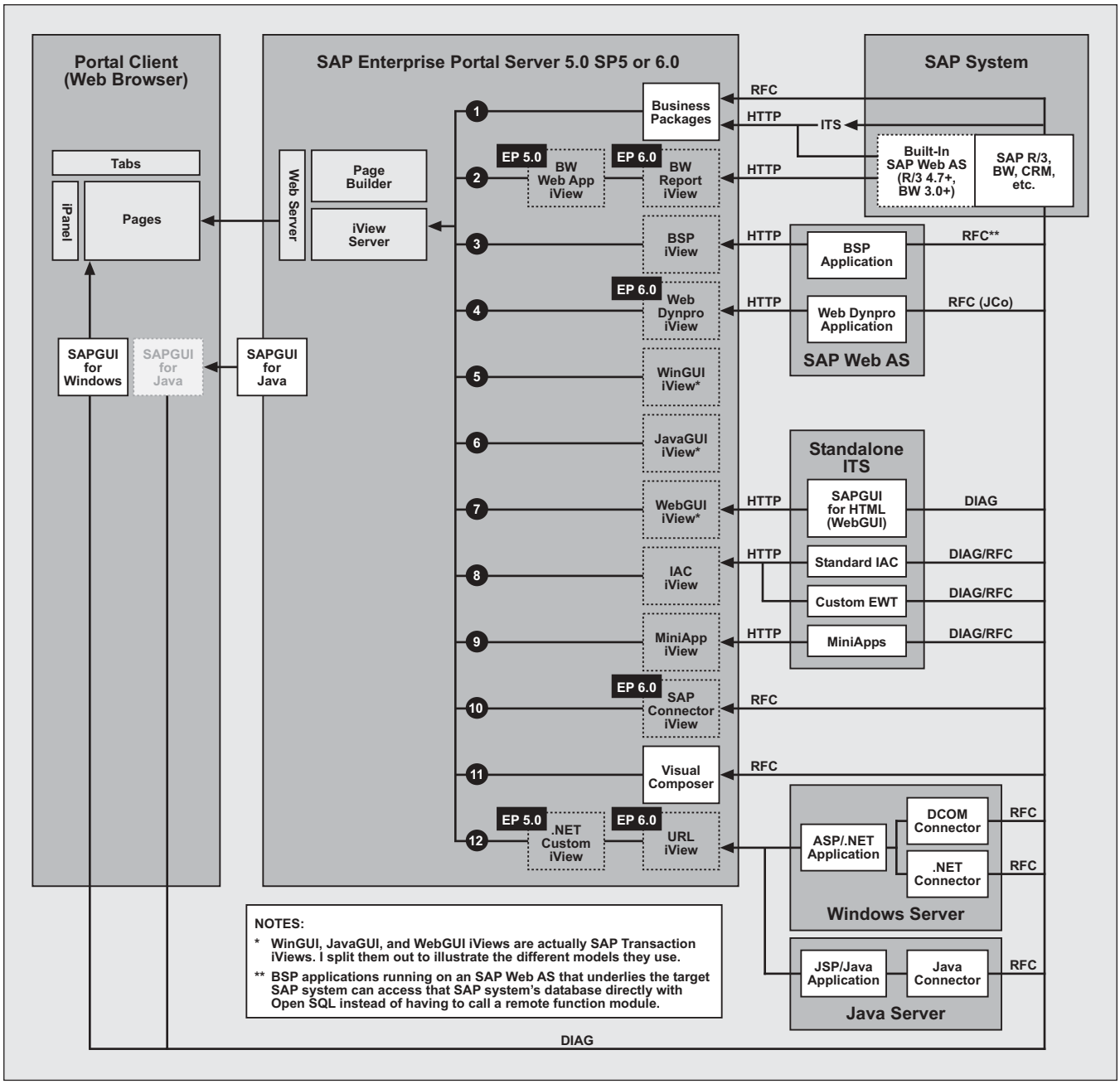


In SAP EP 5.0, system definitions are stored in two files — systems.xml and JCODestinations.xml — which you'll find in the file-system-based portal content directory. To add or maintain the available systems, you'll need to download, edit, and upload these files. For convenience, the portal administration 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.*

Figure 7 Portal Integration Options for Existing SAP Content and Applications



the 12 options in **Figure 7** can be grouped into the following 4 categories:

- Business packages for SAP systems (option ❶)
- Existing Web-based SAP applications (options ❷, ❸, ❹, ❽, ❾, and ⓬)
- SAP transactions and reports (options ❺, ❻, and ❼)
- SAP data exposed by BAPIs or function modules (options ❿ and ⓫)

In the following sections, I will 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.

> ✓ ***Note!***
>
> *To save time and avoid delays, 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 or subsequent sections. Instructions on how to set up these files are included in the download available at* ***www.SAPpro.com****.*

Business Packages for SAP Systems

As I mentioned earlier, **www.iViewStudio.com** should always be one of your first stops when searching for iViews that meet a new content requirement. This is especially true when integrating SAP content, since SAP offers several packages and developing custom applications can be very expensive. As shown in Figure 7 (option ❶), business packages for SAP systems usually connect through one of three methods:

- By calling SAP function modules via Remote Function Calls (RFCs)
- By calling an ITS application or ITS SAPGUI for HTML[19]
- By calling a Web application that runs on an SAP Web AS underlying the SAP system

Some packages use more than one method, like the ones we'll discuss here.

[19] SAPGUI for HTML (WebGUI) is one of SAP's three SAPGUI products (see the sidebar "Choosing the Right SAPGUI" on pages 68-69), and it is a key component of SAP Internet Transaction Server (ITS). All users need to access SAP transactions, reports, etc. via a browser — WebGUI dynamically generates HTML pages that look and function almost as if you were using WinGUI.

> ✓ ***Tip***
>
> *In my experience, most portal teams end up using at least one business package in production. The most popular "generic" business packages are the Communication (for SAP EP 5.0) and Collaboration (for SAP EP 5.0 or 6.0) business packages. Of course, the number of packages you use will depend on your particular requirements.*

There are two particularly useful business packages for SAP system integration you should know about:

- The Employee Self-Service (ESS) package
- The Customer Relationship Management (CRM) package

Available for SAP R/3 4.6B and 4.6C-4.7,[20] the ESS package provides over 70 iViews that employees can use to create, display, and change select portions of their SAP R/3 HR data (see **Figure 8**). The package also includes 7 "worksets" (and corresponding pages) that group the iViews into logical categories for easy deployment. If you don't like the standard pages, you can simply assign the iViews individually to your custom pages.

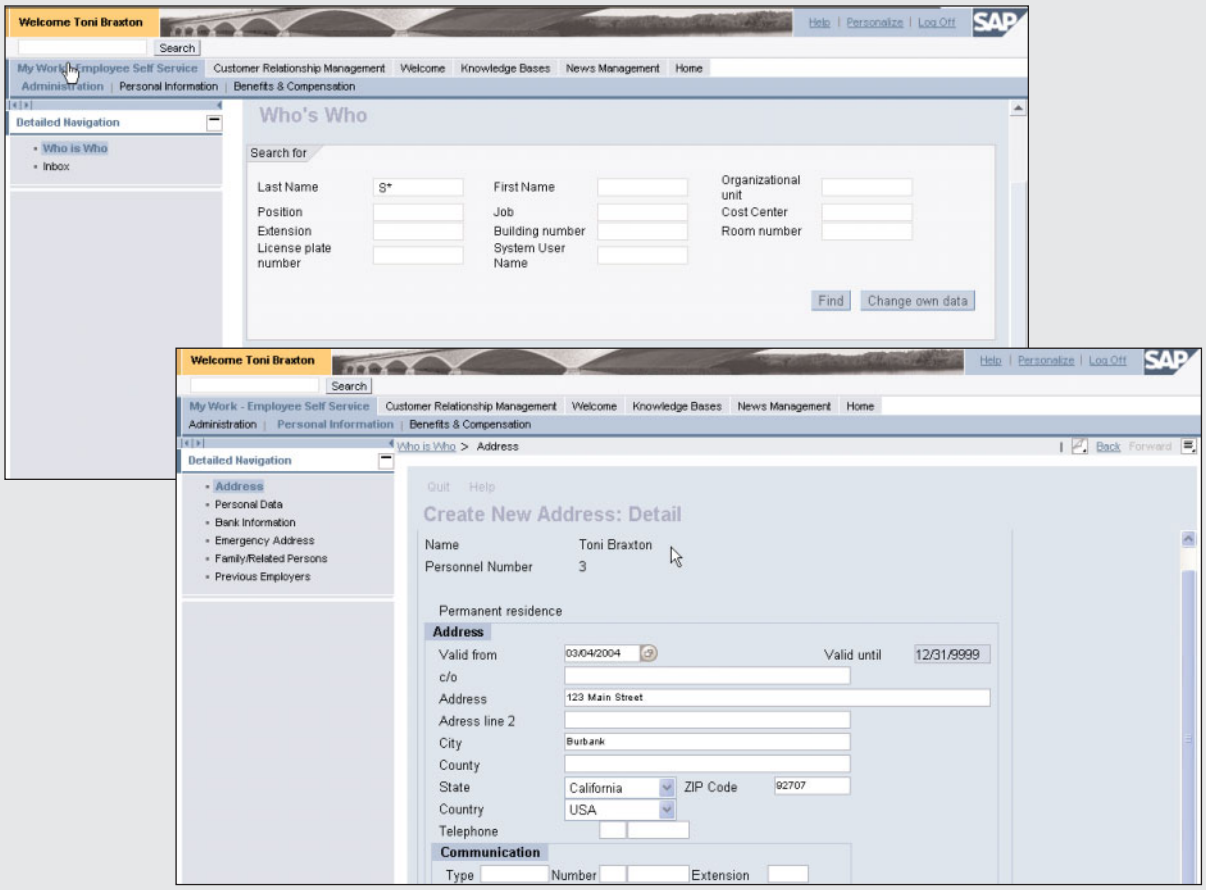
Sounds easy, right? Not so fast. There are two tricky implementation issues.

The first tricky issue is the system requirements for ESS. As mentioned in the documentation, ESS is implemented as a set of ITS Internet Application Components (IACs)[21] that come with your SAP R/3

[20] The business package for ESS 4.6B will only run on SAP EP 5.0. The business package for ESS 4.6C-4.7 will run on both SAP EP 5.0 and 6.0.

[21] Internet Application Components (IACs) are ABAP transactions built specifically for Web use through ITS. ITS dynamically converts each ABAP screen to HTML at runtime with the help of screen-specific HTML templates placed on the ITS server. While you can technically launch them from SAPGUI if you know the transaction code, they're rarely usable, since they usually lack important Dynpro elements like a GUI status.

Figure 8 Example iViews from the ESS Business Package



✓ *Note!*

For more on system requirements for the ESS business package, navigate to its details page at ***www.iViewStudio.com*** *and click on one of the documentation buttons in the upper right-hand corner. For details on installing and configuring the ESS SAP R/3 and ITS backend systems, visit the following areas at* ***http://help.sap.com****:*

- ***For SAP R/3 4.6B****: SAP Library → Cross-Application Components → SAP Internet Applications → Employee Self-Service*
- ***For SAP R/3 4.6C****: SAP Library → Cross-Application Components → Employee Self-Service*

system, so to run ESS you'll need an ITS server running the appropriate ESS HTML templates for your SAP R/3 version. Most of the ESS iViews are ITS IAC and ITS MiniApp iViews that simply house the HTML returned by ITS. A few of the ESS iViews, however, are Java iViews with underlying portal components that run on the portal server. These iViews retrieve and post SAP R/3 data by calling specific function modules that are installed by a release-specific version of the SAP R/3 plug-in[22] (see SAP Note 354134 for details).

The second tricky issue is deploying ESS iViews securely. If your company is like most, you'll want to carefully control which data users can view and maintain. ESS provides two levels of control, and I recommend you use them both. The first level is on the portal server, where you can assign specific ESS worksets or pages to user roles. ESS provides several preconfigured iViews for records, working times, appraisals, and skills that you can use either directly or as a starting point for building custom iViews tailored to your organization. The second level is on the SAP R/3 server, where you can manage SAP user ID profiles so that they only have authorization to run particular ESS transactions and execute certain functions. When ESS is installed in your SAP R/3 system, your security/Basis administrator will be able to assign users to these roles. More details can be found at the **http://help.sap.com** areas referenced in the note on the previous page.

The important thing to understand is that SAP R/3 authorizations *always* dominate — no matter which iViews you assign to a user, he or she must have the necessary SAP R/3 authorizations for the iView to access the needed data, transactions, or functions. If a user has insufficient authorizations, ITS will generate an error message in the iView, or the iView simply won't work as expected. Thus it is critical that you closely align your SAP R/3 and portal authorization or role structures and coordinate changes.

✓ *Tip*

The portal administration tool includes a wizard for importing SAP R/3 roles into your portal. While using this wizard sounds like a good idea, there are three good reasons that you shouldn't use it: (1) duplicating SAP R/3 roles on the portal side yields an excessive number of roles to administer; (2) you'll need to manually track and apply any post-import changes made to the portal roles to your SAP R/3 roles, and vice versa; and (3) many companies want to maintain an alternate or less granular portal role hierarchy — e.g., if portal users can be grouped in a simpler way. A more practical approach from an administrative standpoint is to define a few "wide" ESS roles in your SAP R/3 system, and use portal roles to filter out the iViews a particular group of users should see. This is a slightly less secure approach than a full SAP R/3 role design, because a portal administrator could accidentally assign an iView a user shouldn't have, but it will greatly simplify administration. In all cases, however, user visibility is always restricted to his or her own data.

✓ *Note!*

A final important issue you'll want to resolve before beginning any portal implementation project is whether you'll want to implement single sign-on.[23] Many companies consider HR data so sensitive that they want portal users to log in to SAP R/3 explicitly before accessing their HR data. In this case, ITS administrators are usually also instructed to set the inactivity timeouts for the ESS service session to a much lower value than those on the portal server to further discourage unauthorized access, if a user walks away from his or her computer, for example.

[22] An optional SAP R/3 component installed by a Basis administrator that installs extra function modules, programs, and other components needed for SAP EP, BW, SCM, etc. to exchange data with SAP R/3 and vice versa.

[23] A mechanism whereby the backend system (ITS and SAP R/3 in this case) detects the portal user's ID, maps it to a local user ID (e.g., an SAP user ID), and logs the user on without presenting a logon screen. The backend system trusts that the portal has authenticated the user.

With over 1,000 objects[24] for SAP CRM 4.0, the CRM package is the largest of all the business packages. Its iViews are divided into lots of useful subcomponents like Internet Customer Self-Service (iCSS) that customers can use to manage their own CRM information. Some of the subcomponents, like iCSS, require you to install extra components on your SAP CRM system. Other components retrieve reports from SAP BW. The best way to evaluate each of the subcomponents is to study the CRM package documentation on **www.iViewStudio.com** and pay close attention to the prerequisites for any iViews of interest.

> ✓ ***Tip***
>
> *For more information, consult the CRM business package installation guide (at **http://service.sap.com/crm-inst**, navigate to SAP CRM 4.0 → People-Centric CRM Inst. a. Config. Guides → Administration of the Business Package for SAP CRM 4.0 50.2). This guide is over 200 pages long and provides details on all the system configurations required to enable the iViews to work correctly.*

> ✓ ***Tip***
>
> *The CRM business package was initially developed for SAP EP 5.0, but there is a separate package for 6.0 available for download from **www.iViewStudio.com**. Before importing the package, note any objects in this view and consider moving them to a new folder, because the huge number of objects imported by the CRM package will make the old ones hard to find (or identify). After loading the package into your SAP EP 6.0 portal, you'll find its components (roles, iViews, etc.) under the path Portal Content → migrated content → EP5.*

[24] Roles, pages, iViews, etc.

Business Package Download Restrictions

Many of the business packages on **www.iViewStudio.com** are "restricted." When you register as a user of the site, you will be asked for your SAP Customer and Installation Number, which determines the SAP licenses you have. When a business package is marked as "restricted," your license is checked against the license required for the business package. If your license includes the business package, you can click on the blue "Add to" button to add it to your download basket. If you do not have a valid license, the "Add to" button for the package is disabled (i.e., it is "grayed out").

Some final tips when searching for, installing, and using business packages:

- ☑ When you visit **www.iViewStudio.com**, you'll be asked to log on either as a guest user or as a registered user. You can search the content catalog freely as a guest user, but you'll need to log on as a registered user to download any packages. The registration process checks your license agreement and selectively download-enables packages that are covered by your particular license (see the sidebar above for more details).

- ☑ The best way to search for packages is using iViewStudio's keyword search feature (the bottommost option on the content catalog selection screen). Unfortunately, the search tool does not let you search the packages by SAP EP release. A trick that often works is to do a keyword search on just "EP 5.0" or "EP 6.0." This technique is not foolproof, however, since the SAP EP version is sometimes listed in the associated documentation instead of the searchable package description. Remember also that

some packages listed as 5.0-only will still work on 6.0. See SAP Note 642775 for details on how to find out which version a business package applies to.

☑ Be aware that once you've installed a package, you cannot uninstall it! While the extra iViews, roles, and pages shouldn't hurt anything, more objects mean more clutter. So be prudent in your selections, and consider installing new packages on a sandbox portal first or doing a full-system backup of your portal server and database before installing the package.

Existing Web-Based SAP Applications

The next easiest type of content to integrate is existing Web applications that leverage SAP data. If you have SAP BW 3.0 or higher, SAP ITS, SAP Workplace, or any version of SAP Web AS, chances are you have one or more of the following types of applications:

- **BW Web applications:** BW Web applications are built with the visual Web Application Designer in SAP BW 3.0 and higher (see option ❷ in Figure 7). Web-enabling queries with the Web Application Designer is the better of the two ways to integrate them into your portal, since you have full control over the user interface and the output is native HTML.[25]

  As illustrated in Figure 7, use the BW Web Application Java iView wizard in SAP EP 5.0 or the SAP BW Report iView template in SAP EP 6.0. In both cases, the actual HTML is generated by the SAP Web AS system underlying BW — the portal simply launches the query via HTTP and delivers the unmodified HTML result to the browser.

> ✓ ***Note!***
>
> *Depending on the isolation mode you choose when defining the iView (see the sidebar on the next page), the portal may instead provide the browser with a URL to launch the query itself.*

- **Business Server Pages (BSP) applications:** BSP applications can run on a standalone or embedded[26] SAP Web AS 6.10 or higher server. SAP EP 5.0 offers two Java iView wizards for integrating BSP applications: BSP Application (Standard) and BSP Application (with Alias). If you have SAP EP 5.0, you'll use the "Standard" option most of the time — it generates a generic launch URL with basic customization options (e.g., language, SAP Web AS system, namespace). Use the "with Alias" option when you need finer control over the parameters passed as part of the launch URL (e.g., to pass custom start name and value pairs to your application).

  In SAP EP 6.0, these options are consolidated into a single BSP Application iView template, which provides a similar level of control over parameters. Since they are so similar, I've collectively referenced them as option ❸ in Figure 7.

> ✓ ***Tip***
>
> *If you have SAP EP 6.0 and you experience intermittent problems with some BSP applications, apply the latest patch level to both your J2EE Engine and portal server. See SAP Note 660342 and related notes for details.*

[25] The other, less attractive (but easier) approach is to roll out your Business Explorer (BEx) queries directly to users using SAP EP 5.0's .NET Custom iView or SAP EP 6.0's URL iView template. In this case, the query's Excel spreadsheet comes up in the iView pane. This option will work with all BW versions.

[26] Modern SAP systems like BW 3.0 and higher, SCM 4.0, and CRM 4.0 run on top of an SAP Web AS. You can also set up and run SAP Web AS as a standalone system on which to run custom BSP, Java, or Web Dynpro applications.

Understanding iView Isolation Levels

SAP Enterprise Portal includes a concept called "isolation levels" in SAP EP 5.0 ("isolation modes" in SAP EP 6.0) that is used to control the path content takes from the source system to the browser. Let's take a look at the options available to you in SAP EP 5.0 and 6.0.

SAP Enterprise Portal 5.0

In SAP EP 5.0, most iViews can be assigned a specific isolation level using a dropdown on the Load tab in the iView Editor, which offers the following options:

- ☑ **Simultaneous load:** This is the default option. The text of the option will be either ".NET iviews" or "Java iviews," depending on the type of iView you are creating. In either case, at runtime the portal collects the content for all iViews on the requested portal page and delivers the completed page to the user's browser. This option minimizes network traffic — it delivers pages in one request-response cycle, and users' browsers don't need network access to the backend systems. The main downside to this option is that the entire portal page is refreshed each time a user clicks on a link or button or reloads an iView. This needless refreshing can make the portal seem slow.

- ☑ **Individual load:** This option instructs the portal server to isolate the iView in its own iFrame* within the portal page. The iFrame is loaded individually by the browser using the source system URL assigned to it by the portal server. The source system URL is then automatically embedded within a URL that points back to the portal server, which fetches and returns the content in a separate request from any other "non-isolated" iViews on the page. SAP recommends this option because it consumes less portal resources — e.g., the portal doesn't have to hold 80% of the content waiting for one slow iView to load — and because it lets users see the content more quickly. Also, user actions affect only the iView in which the action occurs, not the entire page. The downside to this option is that some iViews won't work, because the source system URL embedded within the iFrame URL does not work (this often happens with Yahoo! pages). The best approach is to try it and see if it works.

- ☑ **By-pass:** This option tells the browser to bypass the portal server's iView rendering mechanism entirely and get the iView's content directly from the source system. Bypass iViews get loaded into iViews just like individual load iViews, and the iFrame is provided with an unmodified version of the source system URL. This option is a great choice for iViews that don't work with individual load, and for scenarios in which users' browsers will be able to establish a network connection to the source system via HTTP.

For more information, go to the SAP EP 5.0 documentation at **http://help.sap.com/portals** and follow the path *SAP Enterprise Portal Documentation → Developer Guide → Developing iViews → iView Architecture → Isolation Levels*.

---

* An iFrame is an HTML tag. The iFrame element creates an inline frame that contains another document.

(continued on next page)

(continued from previous page)

SAP Enterprise Portal 6.0

In SAP EP 6.0, isolation levels are known as "isolation methods" and have been reorganized. You can set an iView's isolation method by selecting "Load" from the property category menu in the iView Editor. You'll find the following options:

- ☑ **Embedded:** This option corresponds to the SAP EP 5.0 simultaneous load option. iView content is collected, and the finalized page is then presented in the browser. When the user clicks on any link or button, the entire page is reloaded, which can be slow.
- ☑ **Isolated URL:** This option corresponds to the SAP EP 5.0 individual load option. Content is taken directly from the source system and presented in the browser. The benefit is that iView actions don't affect the rest of the page.
- ☑ **Pumped:** This is a useful new option you'll probably want to use extensively. Like with the embedded option, the portal server retrieves the content and downloads the entire page all at once. The difference is that the page is generated with an iFrame for each iView, and JavaScript code then individually writes the content for each iView into its iFrame. The benefit of this option is that the page doesn't need to refresh after each user action (like isolated URLs), but it still routes all traffic through the portal server, so your users only need connectivity to the portal server. This is particularly useful if you're connecting over the Internet — you can provide a secure HTTPS connection to your portal server and keep the backend systems securely behind your firewall.

For more information, go to the SAP EP 6.0 documentation at **http://help.sap.com/portals** and follow the path *SAP Enterprise Portal Documentation → Administration Guide → Portal Platform → Content Administration → Pages → Page Editor → Assembling a Page → Isolation Method of iViews.*

✓ ***Note!***

In addition to tweaking iView isolation levels, SAP recommends a number of different caching mechanisms to improve iView performance. I have found in practice, however, that when performance problems occur, it usually isn't iView caching that makes a noticeable difference, but the performance parameters set in the J2EE Engine, especially the amount of server memory allocated to the J2EE Engine.

- **Web Dynpro applications:** SAP EP 6.0 includes a template for building iViews for Web Dynpro applications (option ❹ in Figure 7). Web Dynpro is a new programming model for developing Web applications. It consists of a runtime environment and a graphical development environment with special Web Dynpro tools that are integrated in the ABAP Workbench. As a NetWeaver product, Web Dynpro is fully integrated with the portal development environment, which means that the Web Dynpro runtime automatically uses the current portal theme. If you start the Web Dynpro application outside the portal, the standard SAP theme is used. Be aware, however, that Web Dynpro applications run only on J2EE 6.30 and above — since the current portal release (SAP EP 6.0 SP2) runs on 6.20, Web Dynpro applications have to be hosted remotely. Further details about Web Dynpro can be found at **http://sdn.sap.com**.

- **ITS applications or MiniApps:** If you're running any ITS IACs like ESS or the old SAP Online Store,[27] or if you developed any custom Easy Web Transactions (EWTs),[28] you can instantly integrate them using the SAP IAC iView wizard (option ❽ in Figure 7). If you're still running SAP Workplace and have any ITS MiniApps you think portal users will find valuable, you can integrate them in place with the SAP MiniApp iView wizard/template (option ❾ in Figure 7).

  I have faced two main challenges when integrating ITS applications. First, ITS applications use one of three logon models:

  - With *explicit* logon, ITS generates a login page that asks the user for an SAP user ID, password, and (optionally) language
  - With *public implicit* logon, the administrator has hard-coded an SAP user ID and password in the application's service file (or the *global.srvc* file), so the application doesn't require the user to identify himself or herself
  - With *private implicit* logon, the model is the same as public implicit logon, except that the application generates its own login screen and validates the user's ID and password against an internal database (usually the standard one maintained with SAP R/3 transaction *SU05*)

  It's important that you identify up front which model your application uses in order to predict whether users will be prompted to log on, even in a single sign-on environment. Nearly all business applications for SAP R/3 users (e.g., ESS) use the explicit logon model. Applications for users without an SAP user ID (e.g., customers or vendors using the old SAP Online Store IAC) typically employ the public or private implicit logon model, or both. SAP Workplace administrators learned several years ago that you can use single sign-on to suppress the logon for the explicit model, that the public implicit logon model doesn't require a bypass, and that the private implicit logon model will *always* force the user to log on. For more information on configuring single sign-on for ITS (and other SAP) applications, visit **http://help.sap.com**.

> ✓ ***Note!***
>
> *SAP has removed vital single sign-on information from the SAP EP 6.0 online help, so use the SAP EP 5.0 help for both versions — single sign-on is essentially the same in both releases. Follow the path SAP Enterprise Portal Documentation → Administration Guide → Security → Single Sign-On → Single Sign-On to SAP Systems.*

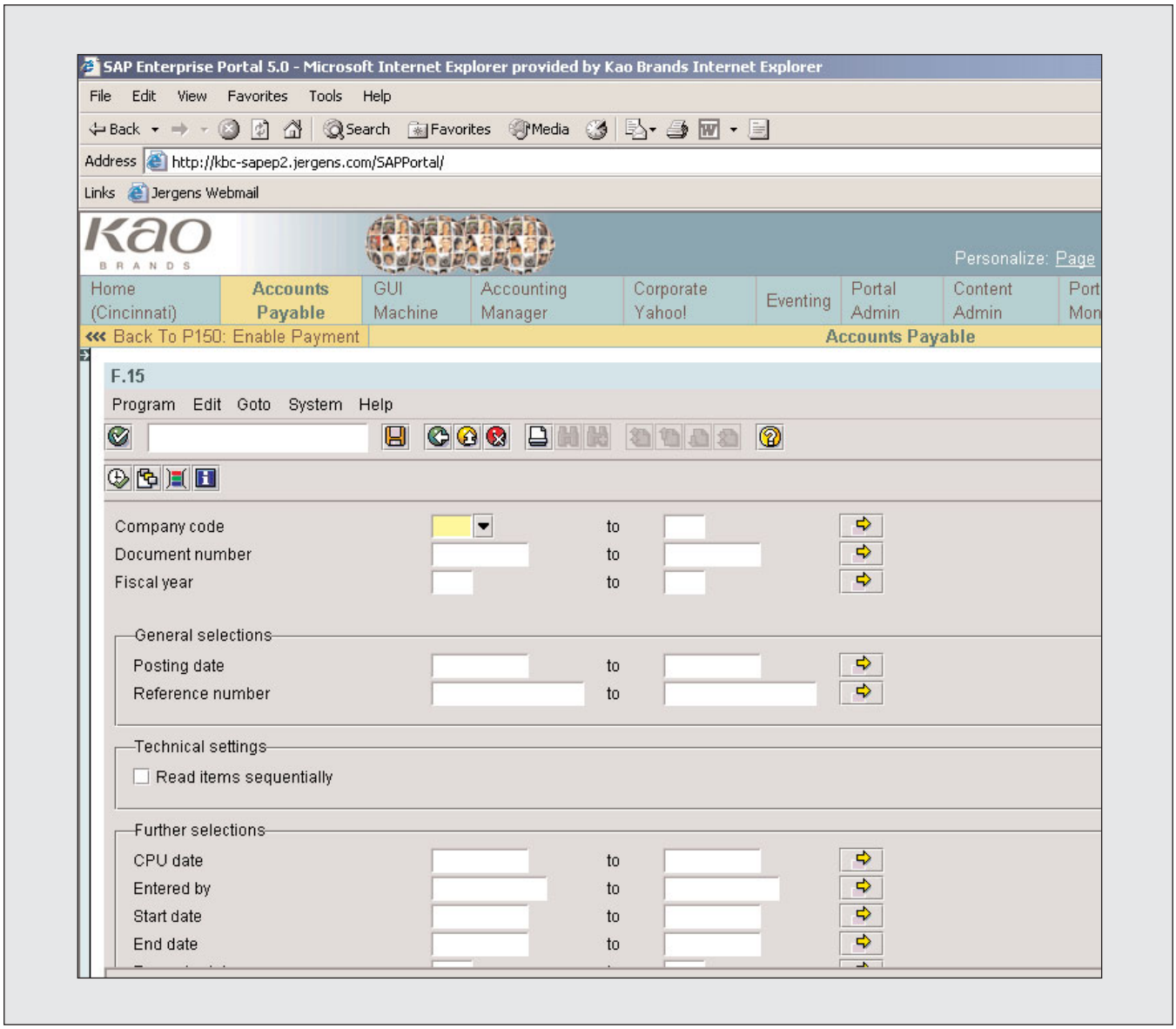
The second challenge is troubleshooting. When an iView does not work, you have three potential points of failure: the SAP system, ITS, or your portal. It's helpful to first remove the portal from the equation by launching the ITS application directly in a browser. If it doesn't work, you know it's either ITS or the SAP system. If you know the SAP transaction code underlying the ITS application, you can then log on and launch the transaction with SAPGUI to see which is the culprit.

- **ASP/.NET applications or Web-based Java applications:** Existing custom ASP/.NET applications that access SAP with the DCOM or .NET Connector, or Web-based Java applications that access SAP with the Java Connector (JCo), are both easily integrated with the SAP EP 5.0 .NET Custom wizard or the SAP EP 6.0 URL

[27] The SAP Online Store has been replaced by SAP Internet Sales. For more on this, see the two-part article series "Quickly and Easily Leverage Your SAP R/3 Data and Processes for Online Sales with SAP Internet Sales (R/3 Edition)" in the July/August and November/December 2003 issues of *SAP Professional Journal*.

[28] EWTs are ABAP transactions built in the normal way but with the Web in mind. The developer takes the additional step of coding an HTML template for each screen, which ITS uses to convert the ABAP screen to HTML at runtime. IACs are simply off-the-shelf EWTs developed and released by SAP.

Figure 9 SAP Transaction iView Displayed with JavaGUI

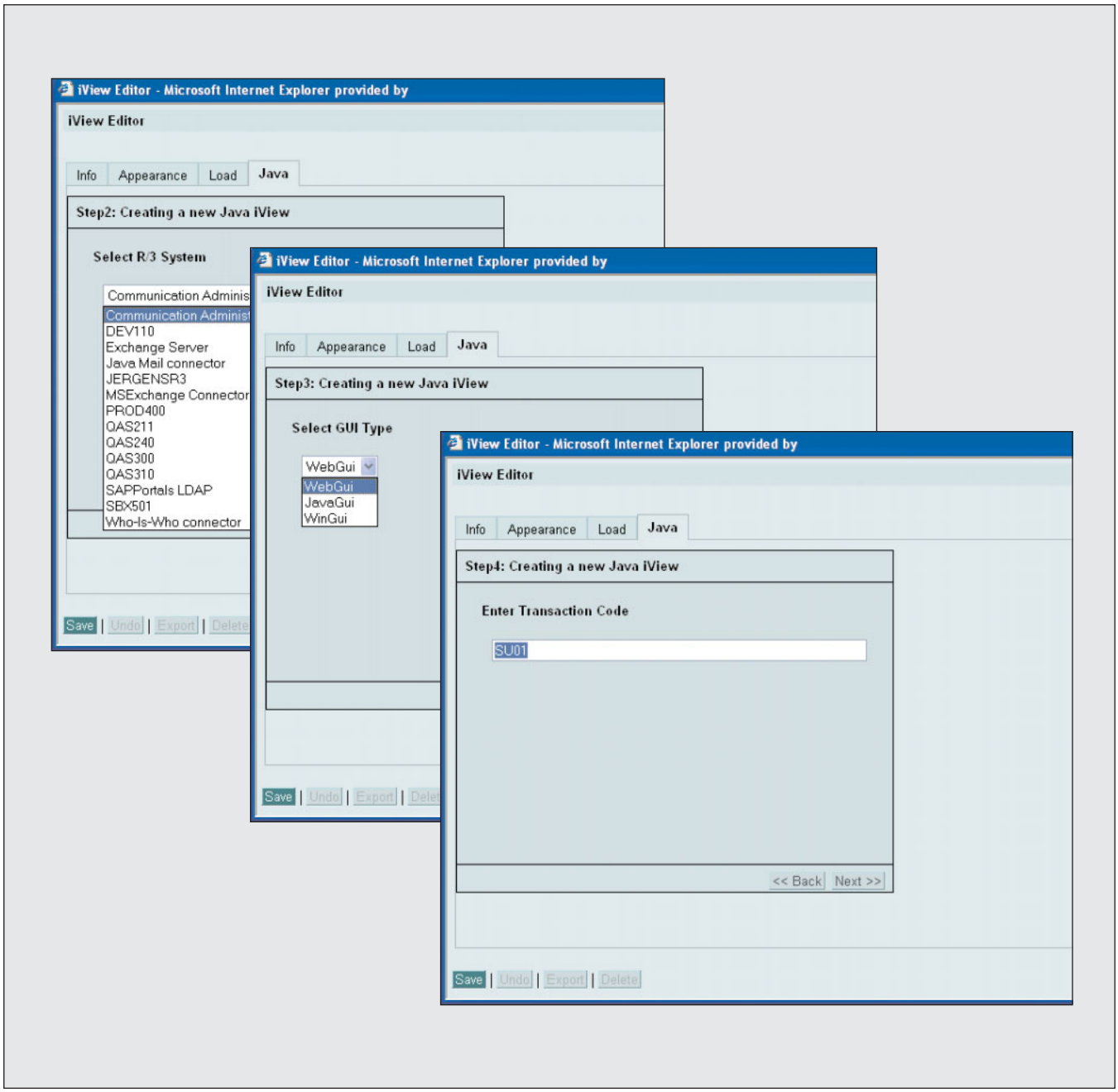


iView template, in the same way you would integrate any other Web resource (see option ❷ in Figure 7). If you find it advantageous, you can host your ASP/.NET applications on SAP EP 5.0, or your Java applications on SAP EP 5.0 and 6.0.

SAP Transactions and Reports

As mentioned earlier, both SAP EP 5.0 and 6.0 include an SAP Transaction wizard you can use to deploy any SAP transaction to portal users via SAPGUI. **Figure 9** shows an example. All you have to do is create an SAP Transaction iView, specify an SAP system and transaction code, and choose which SAPGUI the portal should launch (see the sidebar "Choosing the Right SAPGUI" on pages 68-69): SAPGUI for Windows, SAPGUI for Java, or SAPGUI for HTML (options ❺, ❻, and ❼ in Figure 7). **Figure 10** shows the wizard

Figure 10 Rolling Out an SAP Transaction iView Using SAPGUI



✓ *Note!*

The easiest way to roll out reports is to assign them a unique transaction code. To roll out reports without transaction codes, or queries built with SAP Query or InfoSet Query, roll out the transaction users normally use to access them — e.g., the report tree (SMEN), ABAP/4 Reporting (SA38), ABAP Editor (SE38), or SAP Query (SQ01).

Choosing the Right SAPGUI

SAP offers three GUIs that have a similar look and feel but use very different technologies. Here's how each stacks up (see the table on the next page for a summary):

- **WinGUI** has the largest client footprint (about 80 MB) and can only be used in a Windows environment. It must be loaded onto each client PC. WinGUI offers the most functionality of the three SAPGUIs — e.g., tight integration with Microsoft Office and the Windows operating system — and the best performance, but security is a problem over the Internet, since its communications cannot be secured using SSL.* WinGUI is a good choice when deploying transactions to experienced users on internal networks who are guaranteed to have WinGUI on their desktops.
- **JavaGUI** is essentially a Java version of WinGUI that runs on any PC a Java virtual machine (JVM) can run on (e.g., Macintosh, Unix, and Windows). Its main benefit is that you can install the JavaGUI classes on a Web server instead of on each PC. Upon request from any browser, the classes are downloaded and run on the JVM built into the browser. JavaGUI has a much smaller footprint (16 MB) than WinGUI, but is a bit slower and lacks a few features like Microsoft Office integration. And like WinGUI, its communications cannot be secured using SSL. Thus JavaGUI is a good choice for experienced users on internal networks who don't have WinGUI installed on their PCs or aren't running Windows. You'll find the installers for JavaGUI on Presentation Disk 2 of your SAP installation CDs — there are a number of versions (in subfolders) for specific operating systems.
- **WebGUI** lets you instantly access SAP transactions and reports from any Web browser. It is a standard

* The Secure Sockets Layer (SSL) is the standard way of securing Internet communication. WinGUI offers an alternative, more expensive approach called Secure Network Communication (SNC).

screens for creating an SAP Transaction iView using SAP EP 5.0 SP5 (the 6.0 screens are nearly identical).

Like any powerful feature, however, the key is knowing when and how to use it effectively. Quick, simple, and cheap, this approach is great for experienced SAPGUI users who expect its full functionality and are familiar with its user interface. Providing SAPGUI access through your portal will also promote it as a one-stop workplace for these users (see the sidebar "Should I Launch SAPGUI Inline or Standalone?" on the next page for an important caveat).

On the other hand, using SAPGUI is not the most elegant way to integrate the more generic SAP applications used by novice or non-SAP users. Besides lacking a Web look and feel, SAP R/3 transactions like *IQS23* (PM Create Notification) usually include many fields and functions not needed by a large portion of its audience, so a custom Web application often makes more sense. A less aggressive solution is to create one or more role-specific transaction variants that suppress screens and fields users don't need.

Some other things to keep in mind when deploying SAPGUI iViews:

- ☑ It's almost always best to bring novice users directly into specific business transactions like *VA03* (Display Sales Order), so they don't have to navigate once SAPGUI comes up. More experienced SAP users, however, usually want to navigate freely, so consider rolling out a single, generic SAPGUI iView in addition to or instead of many context-specific ones. The SAP Easy Access menu (transaction *S001*) is a good starting point for end users, as is the ABAP Workbench (transaction *SE80*) for developers.

feature of SAP Internet Transaction Server (ITS)** that dynamically "paints" a Web-based frontend (i.e., an HTML page) that looks almost exactly like WinGUI. All you need is a browser — nothing else runs on the PC. There is also no client footprint, as WebGUI resides entirely on the ITS server. The main downside to WebGUI is that it's a bit slower than WinGUI and JavaGUI. It also lacks certain niceties, such as pop-up windows — e.g., when a user requests an F4 value list, the "pop-up window" loads into the main browser window instead of temporarily replacing the main page. When users click on OK, they're returned to the main screen and the value appears in the target field. Nevertheless, WebGUI is well suited for situations where the SAP system needs to be accessed externally over the Internet, since communication can be easily secured via SSL. For more information on securing WebGUI, download the ITS Administration Guide from **http://service.sap.com**.

Feature	WinGUI	WebGUI	JavaGUI
Client support	PC-only	Internet Explorer or Netscape Navigator	Macintosh, PC, Unix, etc.
Client footprint	Install on client	Zero client footprint (browser only)	Install on PC or Web server for runtime download
Functionality	Full feature set	(Almost) full feature set	(Almost) full feature set
Performance	Fast load time; fast GUI performance	Moderate load time; moderate response time	Local PC: fast load time; moderate response time Web server: moderate load time; moderate response time
Security	SNC, so not well suited for Internet deployment	SSL between browser and ITS server; SNC between ITS and SAP if needed	SNC, so not well suited for Internet deployment

** ITS is a standalone product that is available free of charge to SAP customers from **http://service.sap.com**.

Should I Launch SAPGUI Inline or Standalone?

By default, transactions you deploy via SAP Transaction iViews appear embedded within the browser window. If you're using SAPGUI for Windows (WinGUI) as the GUI, here's a little-known trick you can use to launch SAPGUI natively on the desktop instead:

- In SAP EP 5.0, after creating your iView, bring up its attributes by clicking on the "Edit" link. Next, navigate to the Java tab and scroll down until you locate an option called "GUI starts in place." Set the option to "0."

- In SAP EP 6.0, while creating your SAP Transaction iView (after selecting WinGUI as your GUI type) look for a dropdown called "How to start SAPGUI for Windows." The dropdown offers two options: "SSD" (Structured Storage Document) or "SSF" (SAP Shortcut File). The SSD option starts WinGUI in place in the portal. When WinGUI starts, the portal component com.sap.portal.appintegrator.sap creates a response based on the Microsoft Component Object Model (COM) that contains all the necessary connection data. The SSF option starts WinGUI in a separate window.

These settings cause the portal server to generate a shortcut file when the iView loads instead of its normal

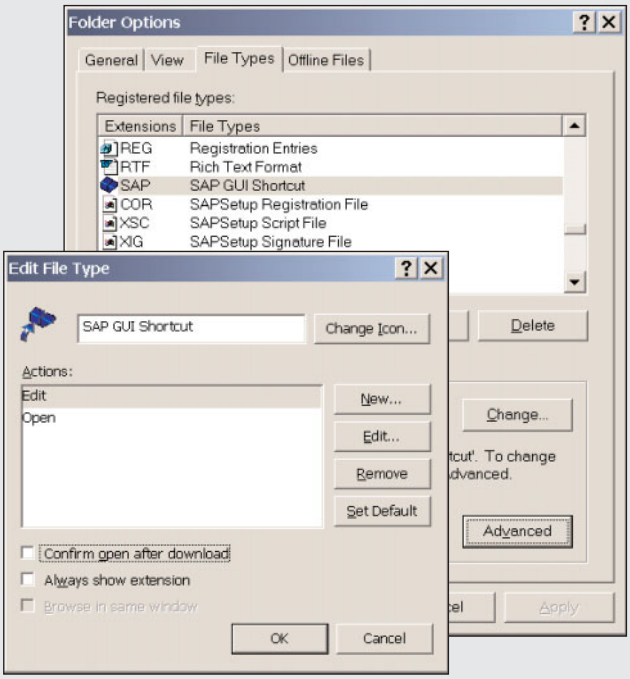
(continued on next page)

(continued from previous page)

behavior (i.e., generating a page with an ActiveX object that launches SAPGUI directly inside the iView pane). The shortcut file contains a link to the target transaction.

This option is great for SAP power users who often get frustrated with the look and feel of SAPGUI appearing inside a browser window. Inexperienced users are often more comfortable with the standard embedded approach, since it preserves the Web context and they can use the browser's back button to return to the previous portal page.

There's one small deployment issue with the shortcut method: for security reasons, when the browser receives the shortcut file, it asks the user if he or she would like to run the shortcut or save it to the file system. Users should select "Run," but this dialog might confuse some. The workaround is to deactivate this security prompt (for SAPGUI shortcut files only) on each user's PC. The screenshots to the right show you how.



- ☑ It is best to remove the iView title if you expect the user will navigate to other transactions. For example, let's say you have an iView containing transaction *VA03*, with the title "Display Sales Order." If the user navigates to the Update Sales Order transaction (*VA02*) within that same iView, "Display Sales Order" will still appear as the title.

- ☑ During testing, optimize the size (depth) of each iView's pane. In SAP EP 5.0, look on the *Appearance* tab in the iView Editor. In SAP EP 6.0, right-click on an existing iView's icon in the portal content directory, select *edit* → *iview*, and choose *Appearance – Size* from the drop-down in the upper right-hand corner. Sizing options include *pixels*, *full page*, or *automatic* (the *automatic* setting sets the page to the size of the first iView/transaction that opens). I recommend that you stick with the *full page* option.

- ☑ If you choose to use WinGUI, each client PC that will access the portal must have it installed. You can set an SAP Transaction iView to download and run JavaGUI, if it is not already installed, by modifying its parameters. WebGUI only requires a Web connection to the ITS server. See the sidebar "Choosing the Right SAPGUI" on pages 68-69 for more considerations.

SAP Data Exposed by BAPIs or Function Modules

To further help portal teams build SAP-based portal applications without programming, SAP developed two tools that automatically generate Web interfaces from BAPIs[29] and other function modules in SAP systems. The first is the SAP Connector iView

[29] Business Application Programming Interfaces (BAPIs) are out-of-the-box function modules for reading and posting SAP business data from or to SAP systems. Unlike most function modules, they adhere to strict naming conventions and are guaranteed to be backward-compatible when you upgrade your SAP system.

> ✓ ***Tip***
>
> *If you want to run more than one BAPI, or need to customize any validations or other processing, consider writing a custom wrapper function module that invokes the others as needed. Don't forget to RFC-enable your function module, however! You'll find this setting as a radio button on the function module's attributes screen in transaction SE37 (ABAP Function Modules).*

template built into SAP EP 6.0 (option ❿ in Figure 7). The template is quite simple — you just specify an SAP system, the BAPI or function module you want to use, and the input and output fields you want the interface to include. The iView automatically generates a rudimentary Web interface in the style of the current portal theme.

This is an excellent approach when a *single* BAPI or existing function module provides all of the functionality you need and you can live with the standard interface it generates. Note that, as with any iView that accesses SAP data, the user will be required to log on unless you've implemented single sign-on. Users must have an SAP user ID to access the application.

The second tool is Visual Composer[30] (option ⓫ in Figure 7), which is available as an add-on for SAP EP 5.0 and 6.0 systems *running on Microsoft SQL Server only* (or that have a Microsoft SQL Server available). Visual Composer goes beyond the SAP Connector wizard with a Web-based WYSIWYG tool for designing customized user interfaces.

Visual Composer has two parts:

- The Storyboard, which is the graphical development environment
- Add-ons, which provide the Java classes required to run iViews developed with Visual Composer

During installation of the Storyboard, a database is built into Microsoft SQL Server that stores any parameter values you enter when building Visual Composer iViews. Visual Composer itself is Web-based and can be called from any browser via the URL *http://<myportalserver>/GMServer*.[31] There is no way to launch it from within the portal administration tool. You'll be asked to log in as a portal administrator.

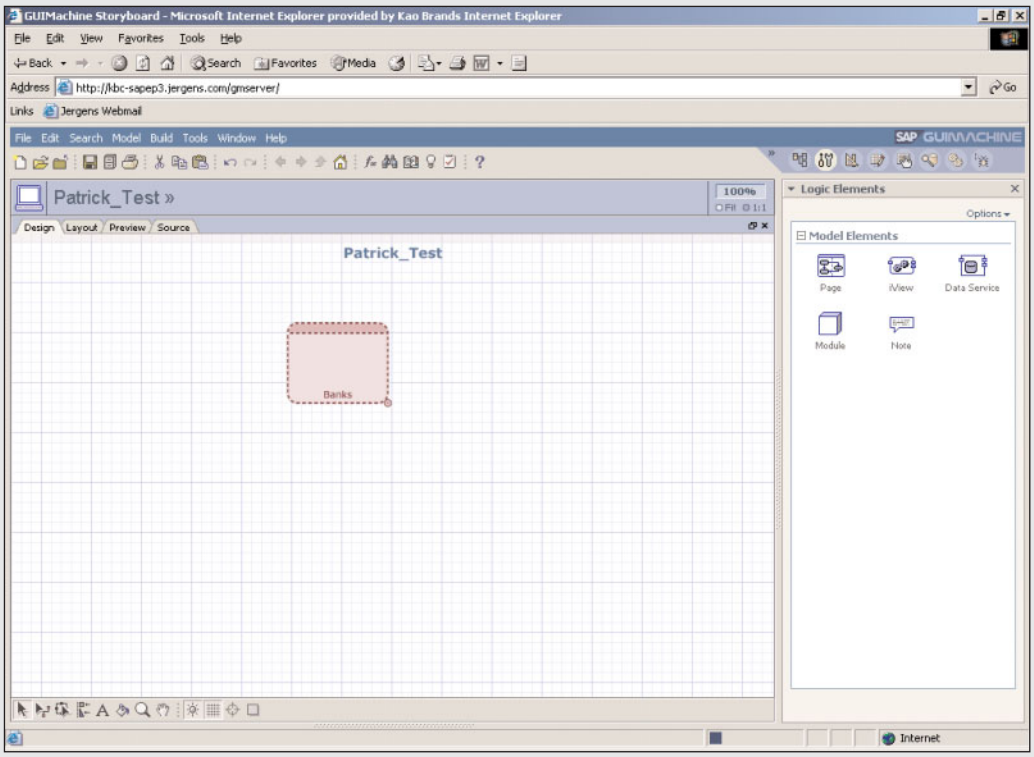
> ✓ ***Tip***
>
> *To run a Visual Composer-developed iView on a portal server, you only need to install the add-ons. Do not install the full Visual Composer on your production portal. Instead, build your Visual Composer iViews on a development system and migrate them to production (using the export and import features in the portal administration tool).*

> ✓ ***Tip***
>
> *If you are using a proxy server and get the message "GM Server not found," check your browser's proxy settings. I once spent half a day uninstalling and reinstalling Visual Composer just to find that someone had changed the proxy settings on the PC that I was using.*

[30] Formerly known as the "GUI Machine," which you'll see in the title bar of the screenshots used to illustrate defining iViews with Visual Composer.

[31] While the tool's name has changed for marketing purposes, you'll see references to "GUI Machine" or "GM" repeatedly when using or administering the tool.

Figure 11 The Visual Composer Design Palette with a New iView



To define an iView with Visual Composer, begin by dragging a new iView from the *Logic Elements* pane on the right onto the design palette. In **Figure 11,** I've dragged an input form onto the palette and linked it to the iView by clicking on the little circles on each element.

Next, as shown in **Figure 12**, I specify on the right that upon submission of the form, the iView should call BAPI_BANK_GETLIST, and I add a grid view (table) to the iView by right-clicking on the iView and selecting *Add Grid View*.

Figure 13 shows the layout palette that allows us to tweak interface elements like field labels, sizes, and positions using the toolbar at the bottom of the screen. In the *Code Compiler* pane on the right, you can see the available options for executing the compile and deploying the objects to your portal server. If the tool reports that the objects were deployed successfully, the iView will be added to your list of iViews in the portal administration tool, and you can add it to a portal page, test it, and use it just like any other iView.

✓ *Tip*

Visual Composer is still a beta product, so it has a few shortcomings — e.g., the names of the columns in grid views are dependent on the SAP R/3 field names, which we know are not always meaningful, even if you speak German. To work around this limitation, make a copy of the table and rename the fields. I know this is not ideal, but the speed with which iViews can be created with Visual Composer usually makes the extra step worthwhile. I trust many of the limitations will be overcome with the production release of Visual Composer.

Figure 12 Linking an Input Form to a BAPI and Adding a Grid View to the Interface

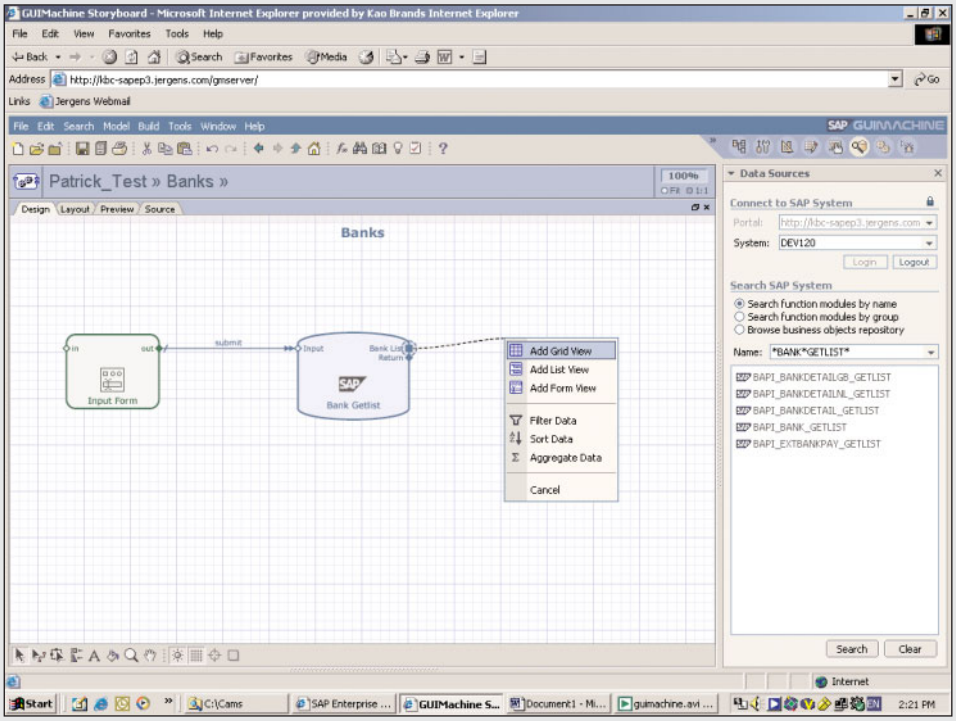


Figure 13 Customizing Interface Elements and Deploying the iView

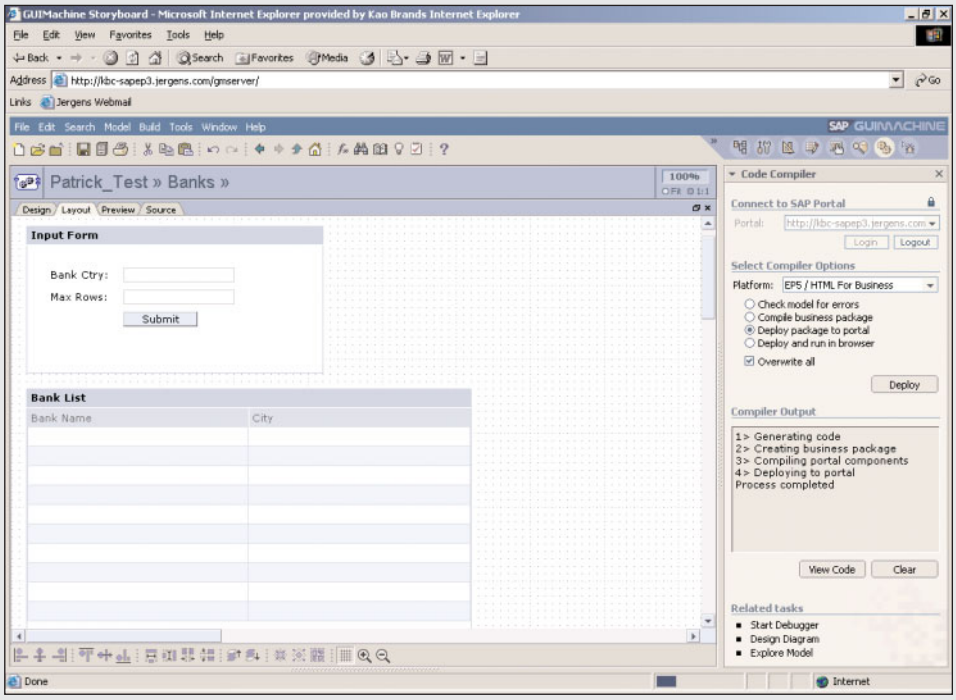


Figure 14 A Visual Composer iView on a Portal Page

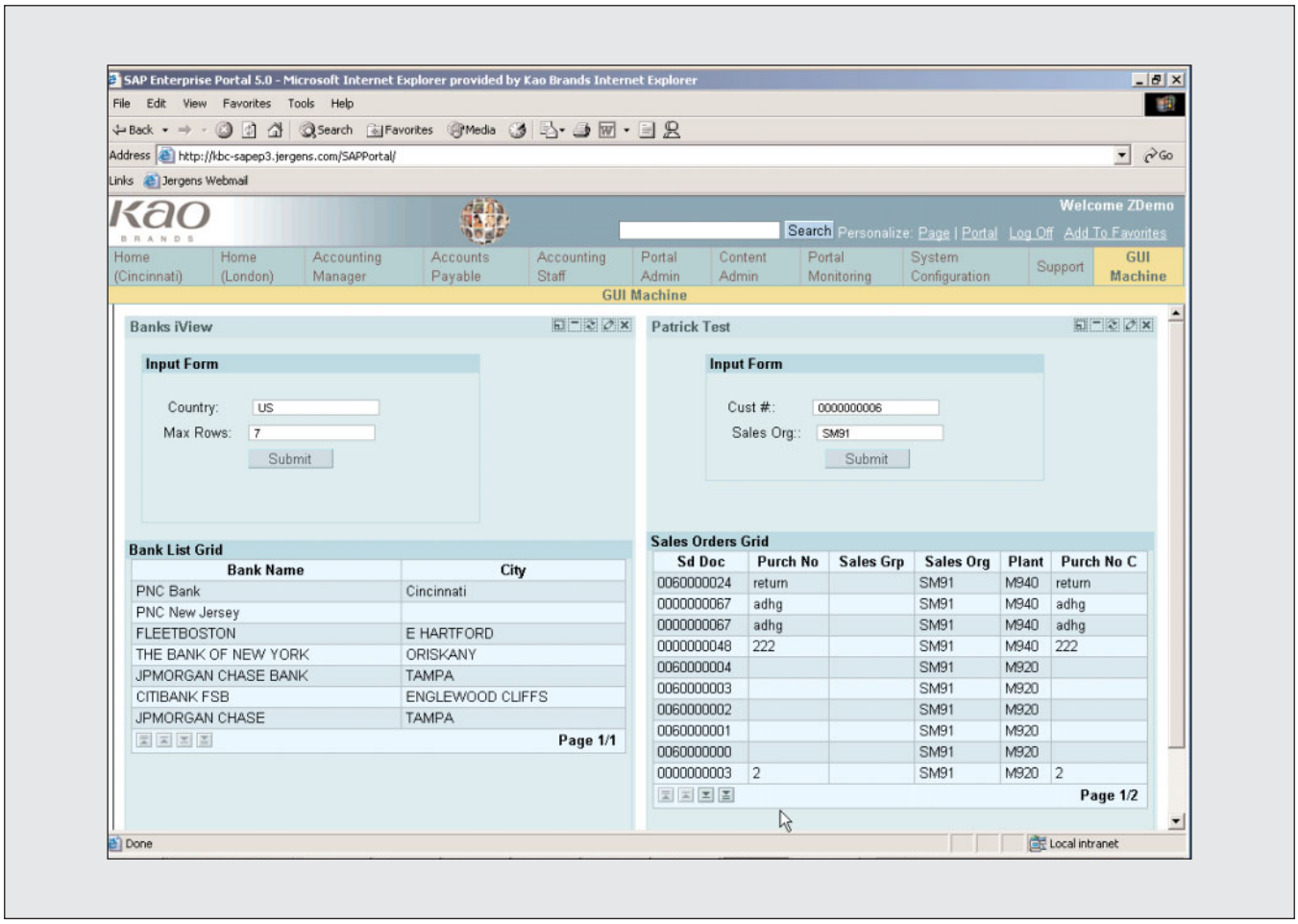


Figure 14 shows the completed example iView at runtime when it has been added to a portal page.

This concludes our tour of the content integration options available to you in SAP Enterprise Portal 5.0 and 6.0. In the following section, I will leave you with some tips that will ensure your own content integration project is a successful one.

Final Tips When Integrating SAP Content into Your Portal

Here are some final things to keep in mind when you set off to start integrating your own SAP content and applications into a portal:

- ☑ WinGUI is by far the quickest way to give your internal users (with WinGUI on their desktops) access to SAP transactions. If users don't have WinGUI installed, and you have an ITS server available, WebGUI will be your next fastest option.

- ☑ Don't forget that business packages usually require additional components — simply installing them won't give you immediate access to underlying SAP systems! The ESS business package, for example, won't work until you have ITS set up and working properly with the ESS

templates, the ESS transactions in the underlying SAP R/3 system configured, and the connections between the portal, ITS, and SAP R/3 set up.

☑ Try some of SAP EP's GUI-based development tools, like the Visual Composer add-on and the SAP Connector and JDBC Database iView wizards. Getting a feel for what they can and can't do early on will help you better estimate how quickly a particular business requirement can be met.

☑ Try to put no more than 8 iViews on a single portal page, and avoid placing iViews with runtimes over 3-5 seconds on users' start pages — they will need to return to this page often and need it to load within 4 seconds. Instead, place links on start pages that users can use to selectively launch iViews with longer runtimes in their own window.

Also, try to heed the "rule of three": content should ideally only be three clicks away (three levels down) from the user's start page. Any more, and users find it difficult to locate content and can get lost in the site.

☑ If you are providing users who are accessing the portal over the Internet with access to SAP transactions, consider securing the connection. The easiest way to do this is with SSL (HTTPS), which is only available for SAPGUI for HTML (WebGUI). For more information, see the ITS Administration Guide (available for download at **http://service.sap.com**).

☑ Because SAP Transaction iViews require significant real estate, don't deploy more than one per page, and try to ensure users only encounter one type of SAPGUI (e.g., WinGUI, JavaGUI, or WebGUI). It is often strategic, however, to choose a different GUI for different user groups — e.g., SAPGUI for users accessing the portal from internal workstations, and WebGUI for remote users who don't have WinGUI on their desktops and are coming in via a virtual private network (VPN).

☑ Remember that all applications that access an SAP (or other backend) system will have to log on to that system with an SAP user ID and password defined on that system. The Web user ID for logging on to the portal is entirely separate.

Your options are:

1. Let the underlying system or application prompt the user for a backend system user ID and password.

2. Implement single sign-on, where the portal server maps the user's portal ID to a backend system ID and essentially logs on for him or her.

3. Hard-code a user ID and password into the underlying application.

> **Warning!**
>
> *The last option — hard-coding a user ID and password — is not appropriate for 99% of business applications, since they require user-level identification, logging, and access control. In many cases, it's a violation of your SAP contract, particularly if you're using the technique to avoid acquiring an adequate number of licenses. SAP has built this capability into ITS and SAP Web AS, however, to enable customers to write public applications that don't require a user logon. Before building these types of applications, consult your SAP licensing representative.*

Conclusion

As you can see, SAP Enterprise Portal offers a powerful suite of wizards, templates, visual design tools, and standard business content you can leverage to integrate content from your SAP, non-SAP, and

Web systems. This article has hopefully armed you with the information you need to choose the best options for rolling out SAP content and applications to your own users, and helped you understand the various roles that components like SAP Web Application Server, SAP Internet Transaction Server, and business packages play in your SAP systems. Keep this information in mind when planning upgrades and your future SAP landscape — the speed with which you can deploy powerful, elegant portal solutions in an optimized environment will save you hours of development time and win you countless fans among your end users.

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.

