



## **Integrating Microsoft® Office SharePoint® Server 2007 and SAP**

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

All other trademarks are property of their respective owners.

# Integrating Microsoft Office SharePoint Server 2007 and SAP

Microsoft Corporation

June 2007

**Summary:** This white paper describes the innovations in Microsoft Office SharePoint Server 2007 that make interoperability with SAP easier than ever. The paper explains the general business and technical options customers gain who want to bring the power of their SAP assets into the tools and places where information workers live. It then shows how SharePoint Server 2007 addresses those challenges, and, finally, describes various interoperability options, from simply displaying SAP information in a portal page to creating complex SharePoint-centered business process solutions that incorporate SAP data.

# Contents

Executive Summary	1
Information Workers need SAP	1
Integration Can Now Be Achieved	1
Introduction for Technical Decision Makers and Architects	2
The Value of Interoperating SAP with SharePoint Server 2007	3
Understanding the Challenges to Integration	4
Challenges in Communications	5
Challenges in Identity and Connection Management	5
Challenges in Composition	5
Challenges in Resourcing	6
Platform Capabilities	6
Business Data Catalog	7
Workflow	11
ASP .NET 2.0 Web Part control set	11
Forms Services	11
Excel Services	12
Single Sign On	12
Integration Options and Best Practices	13
Page Redisplay	14
Rich Collaborative Solutions through Integration with SAP Services	16
Business Intelligence Using SAP Data	18
Conclusion	19
Call to Action	20

# Executive Summary

Presently, the majority of people in an organization do not take advantage of the tremendous investment in information and codified knowledge contained within their line of business (LOB) systems, such as SAP. The lack of integration between their productivity tools and the LOB systems has made it difficult for information workers to get to the data they need.

Microsoft is committed to empowering information workers with simplified access to key business data and processes through the Microsoft Office System. This paper describes how organizations can extend the benefits of SAP to information workers through integration with Microsoft Office SharePoint Server 2007.

## Information Workers need SAP

SAP customers value and rely on SAP for enterprise resource planning in finance, operations, human resources, and other activities and most also rely on Microsoft® productivity tools. The problem is the structured and highly valued content that is maintained so well in SAP is not available to information workers within their familiar unstructured context (creating a document or using a Web portal, for example). This is not the result of a flaw in SAP; rather it results from increasing demands on information workers to provide the connections between the two worlds of structured and unstructured content.

For example, experience shows that a typical business process such as creating a customer quote involves SAP for about 15 per cent of the activities, while the remaining 85 per cent of activities, such as collaborating on documents, communicating with subject matter experts, and analyzing information, occur outside of and disconnected from SAP information and processes. In brief, the work of information workers spans across structured and unstructured worlds, but their tools do not.

## Integration Can Now Be Achieved

This problem is well-known. Indeed, the approaches to integrating information are also well-known. They just haven't been practical or cost-effective because they traditionally required intensive custom-coded applications. One the goal of this paper is simply to demonstrate that the equation has changed. Specifically, innovations in Microsoft® Office SharePoint® Server 2007 make it well within the grasp of SAP customers to integrate SAP assets into core information worker activities that occur within SharePoint-based productivity tools, which represent a major part of the information worker's usual work context.

You'll note that this paper does not discuss the complementary integration possibilities in Office 2007 desktop productivity applications, which, together with SharePoint Server, comprise most of the information worker's context. Another paper, *Building Office Business Applications: A new breed of business applications built on the 2007 Microsoft Office system*, at <http://go.microsoft.com/?linkid=5078276>, takes a broader, futures-oriented perspective of line of business integration with Office System generally, including such subjects as direct integration into Microsoft Word 2007.

Rather, this paper is focused on how new platform capabilities allow customers to make *SharePoint Server* a dynamic consumer of SAP services and processes, either directly or through a Service Oriented Architecture layer, all with little to no code, in ways that bring the unique value of SAP directly into core unstructured activities of information workers. Combined, SAP and SharePoint benefits information worker productivity in several ways:

- Puts tools for creating, deploying, and modifying processes in the hands of business users so they can rapidly build their own processes that augment SAP business applications.
- Integrates tools for ad hoc communication, collaboration, and information management into SAP's structured work environment, empowering people to solve problems that fall outside the specifications of existing processes.
- Enables people and organizations to easily design and deploy electronic forms, automating data capture and providing real-time visibility using familiar user interfaces.

## Introduction for Technical Decision Makers and Architects

This paper is oriented toward Technical Decision Makers (TDMs) and architects and is meant to provide a quick dive into SharePoint Server technologies, design concepts, and best practices, with the primary goal of driving home the realization that integration truly is possible and practicable in ways that it has not been in the past.

The section "The Value of Integrating SAP into SharePoint Server 2007" offers a more detailed scenario-based example of the value of integrating SAP and SharePoint Server 2007 and begins to show the platform components and technologies within SharePoint Server that make integration possible with minimal code. The section "Understanding the Challenges to Integration" offers a brief definition of the generic challenges to any integration project. The section "SharePoint Server 2007 Platform Capabilities" continues by describing the key technologies in SharePoint Server 2007, such as the Business Data Catalog, and shows how they help overcome the challenges to integration.

The last section, "Integration Options and Best Practices", offers guidance and examples of the three basic methods of integration, from simply displaying *views* of SAP information inside SharePoint pages, to dynamically integrating with SAP services to bring *information* into SharePoint activities and composite applications, to surfacing SAP data (without using services) for the purpose of doing analysis and business intelligence.

In the end, a common theme will emerge: Microsoft has worked very hard to create infrastructure capabilities within SharePoint Server that remove the need for much of the custom code that used to be required for integration. As a result, the real work is now in *declaring* and defining how SAP information ought to exist in SharePoint Server, and not in *coding*, deploying, maintaining, and monitoring custom components.

# The Value of Interoperating SAP with SharePoint Server 2007

SAP customers acknowledge that putting SAP into their information workers' natural habitat is a good thing. By connecting SAP to Office, the structured processes in SAP, with their transactional strengths, are enriched with flexibility, speed, and responsiveness. Best of all, information workers enjoy a consistent user experience across their activities.

A number of product-specific integration options exist for any of several desktop and server components of the 2007 Microsoft Office system, but this paper will focus specifically on SAP integration with SharePoint Server. Let us consider a common business process and look at the sequence of events and how systems are used in a non-integrated environment.

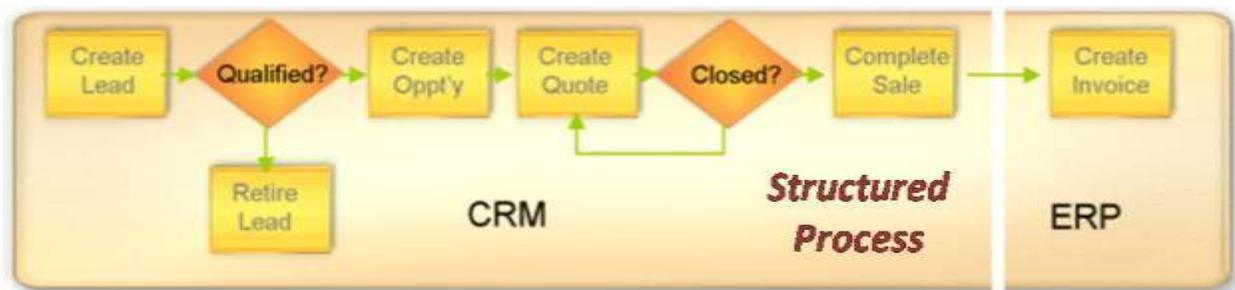


Figure 1: Business process flowchart

Figure 1 shows the typical business sales cycle from start to finish. A sales person has a contact lead and wants to convert that into an actual order. In the perfect scenario, the sales person simply creates the lead, decides when it is qualified, creates the opportunity, performs a quote, closes the quote, completes the sale, and finally invoices the customer, with all activities taking place within the structured business processes that involves SAP and, perhaps, other systems as well.

However, in reality, the process really looks like Figure 2.

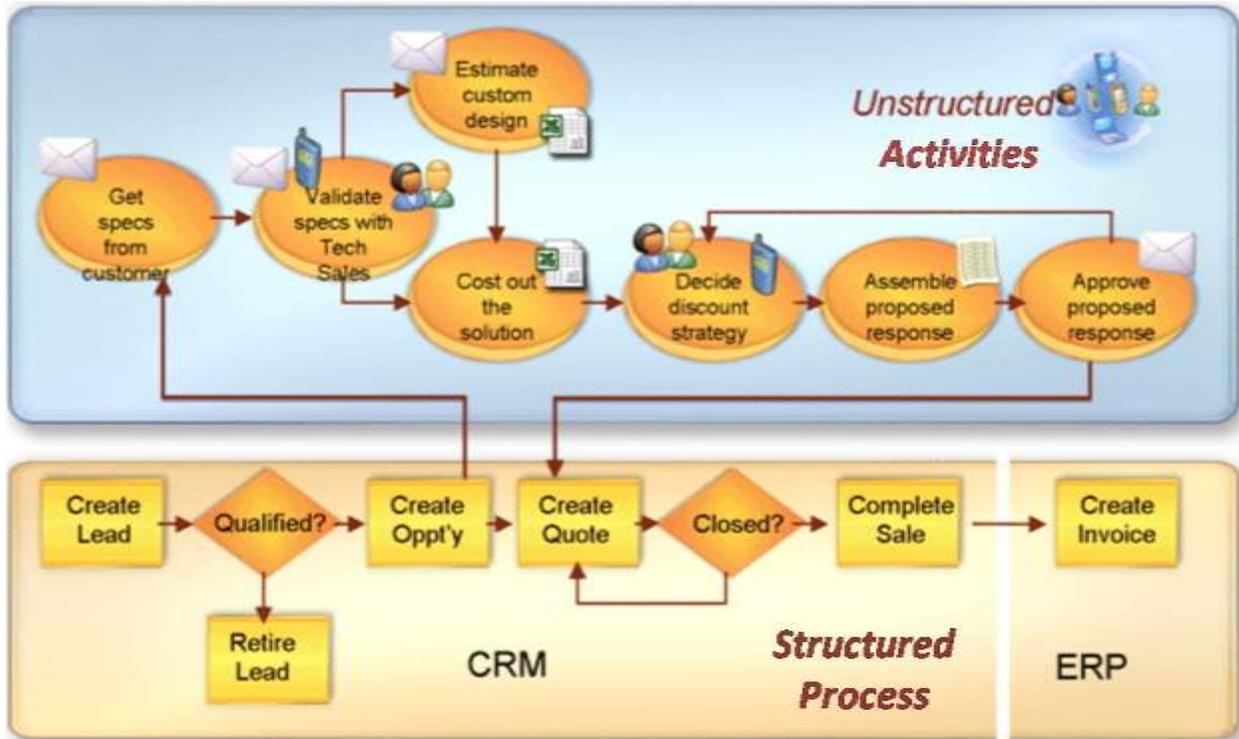


Figure 2: Business processes in the real world

To create the opportunity, someone has to obtain specs from the customer to understand what the customer really wants. They have to decide if this opportunity is valid, and if it is something that can be feasibly delivered. They must cost the solution (probably using Microsoft Office Excel®), and this goes on. In other words, most of this business process consists of unstructured activities, which are not controlled by or accomplished within the back-end system.

By integrating SAP into SharePoint Server in the ways that this paper describes, the unstructured activities in this business process are enriched because the data that was once locked in the back-end is now available at every step. As you'll see later, this is true even with document-based activities (such as costing out the solution in Excel). Further, because integration makes it possible for users to work in a single context, rather than switching tools, they are more productive and the business process cycle itself can be more efficient.

## Understanding the Challenges to Integration

One reality that has not changed is that there are still four basic components to any interoperability scheme:

1. **Communication** – You must be able to connect and communicate with the foreign system, either through code or through infrastructure
2. **Identity Management** – You must have some way of authenticating users against the foreign system

3. **Composition** – You must be able to assemble together discrete pieces of data (from tables, function calls, etc.) into an entity that is meaningful and useful to business users.
4. **Resources** – You must have the developers with the required skills and the bandwidth to write, test and maintain the interoperability scheme.

There are challenges in all four of these activities, regardless of what you are connecting.

## Challenges in Communications

SAP GUI and the SAP NetWeaver Portal are able to communicate with and manipulate SAP ERP Central Component (ECC) information because SAP *wrote code* to enable these capabilities. Likewise, in order to connect a non-SAP front end system to SAP, somebody needs to write code.

Fortunately, the advent of the SOA, where much of the custom code for connecting and manipulating data is moved to another layer, makes integration a more reasonable project. In the “SharePoint Server 2007 Platform Capabilities” section, we’ll see how SharePoint Server 2007 complements the SOA approach and makes solving the communications problem even easier. In the “Integration Options and Best Practices” section, we’ll see how some of the simplest integration techniques don’t need to use an SOA, while the most complex approaches can use it to create composite applications with minimal code.

## Challenges in Identity and Connection Management

Identity and connection management bring additional complications. First, most customers will need to solve the problem of how to manage a user’s multiple accounts and how to pass the proper credentials to the foreign system. Typically, this means either a security authority to which both the portal and the foreign system jointly defer, or a secure means by which the portal can learn and offer foreign system-specific user credentials, acting on their behalf.

Second, even when the above can be achieved, there is still a problem if handling such identity management requires extra coding on the part of component (e.g., Web Part) developers. Identity management needs to be handled through infrastructure (object brokers and other middleware), not by custom application code.

Third, auditing, connection management, and other housekeeping activities, all need to be handled, which means additional code. All of these activities also benefit from service orientation, but SharePoint Server also provides techniques such as single sign-on for avoiding custom code.

## Challenges in Composition

From a bits and bytes perspective, communication and identity/connection management are really all that is required for data to flow, but for people to actually get value out of the information that is being presented there needs to be composition. When querying data from a simple database, composition can often be achieved by joining tables in a single SQL statement. When retrieving data through Web services, however, matters are rarely this simple.

For example, it is possible to create a connection to an SAP ECC system and retrieve various pieces of data about a customer record using Business Application Programming Interfaces (BAPIs) and Remote Function Calls (RFCs). One RFC might return the customer name, while another might return the address, and so on. However, without some composition of these various units of information into a coherent, composite entity, it would be difficult to experience the value that integration brings.

Moreover, there is no guarantee that two Web service method calls will return consistent results. It is quite possible that a method that retrieves a single item will retrieve a different set of properties than a method designed to retrieve a list of such items. What data a Web service retrieves is in the hands of the developer who wrote it. Robust composition services can compensate for this.

Going beyond simple composition of related pieces of information *among themselves*, it is also important to provide composition at the portal level to: blend the portal UI and the Business Application UI; combine the data with portal-hosted collaborative processes; and to provide a means for data visualization inside the portal.

## **Challenges in Resourcing**

While resource challenges are certainly part of the other three challenges, they are really their own category as well. Successful interoperability requires skilled developers at all levels to address problems in security, maintenance, deployment, and so on for at least two systems.

For the remainder of this paper, then, the focus will be on how SharePoint Server 2007 addresses these four challenges to make integration with SAP easier and on specific approaches and best practices for achieving specific integration needs.

## **Platform Capabilities**

SharePoint Server 2007 enables better business insight, accelerates shared business processes, and provides comprehensive content management and enterprise search. These business productivity increases are accomplished through key platform capabilities including the Business Data Catalog, support for Web services in general, personalization, workflow, and electronic forms capabilities. Figure 3 shows how these core underlying innovations in SharePoint Server 2007 enable the four central business capabilities of the 2007 Office System: Unified Communications and Collaboration, Business Intelligence, Enterprise Content Management and Enterprise Search.

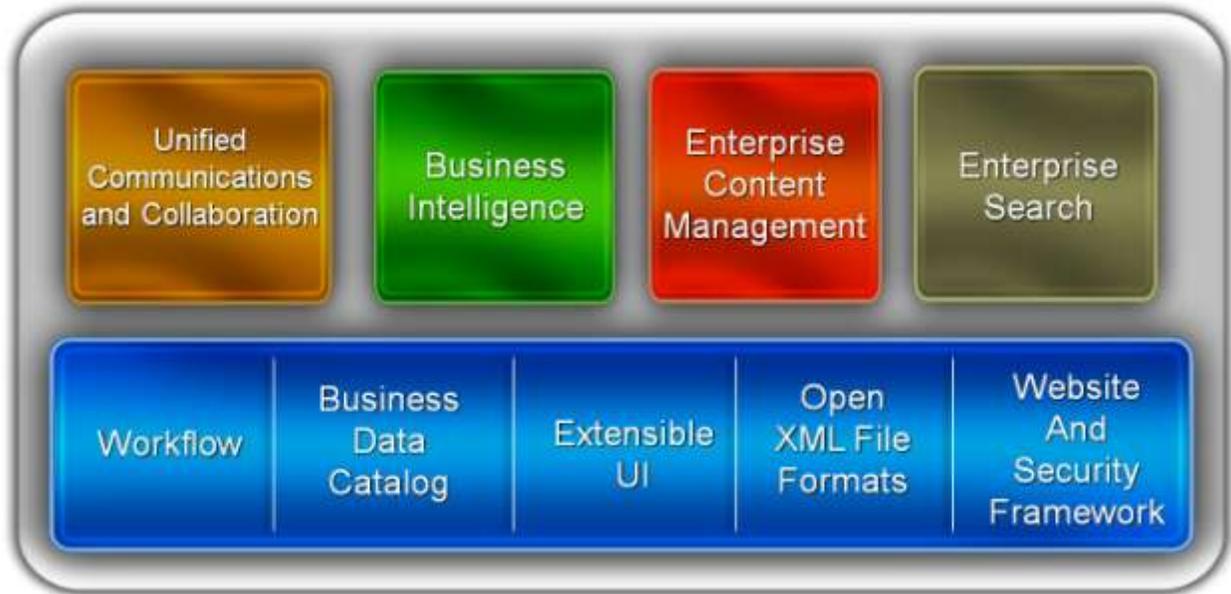


Figure 3: SharePoint Server 2007 - A business portal platform

Together, SharePoint Server 2007 is a true business portal platform that provides all the infrastructure needed for implementing coherent business processes in software that integrate with business systems like SAP, in ways that dramatically reduce the pain of the four challenges to integration described above. As you read about specific technologies in the sections below, keep in mind that a important design goal for SharePoint Server 2007 is to do at the portal infrastructure level what the SOA approach has done for connecting systems in general; SharePoint Server has moved the code required for things like communication and identity/connection management away from individual front-end applications and into infrastructure. Microsoft has written the code once, so customers can achieve results by only defining and configuring connections, UI, and so on.

## Business Data Catalog

The Business Data Catalog is a new business integration feature in SharePoint Server 2007. It is a shared service and it enables SharePoint Server to surface business data from back-end server applications without requiring custom code for managing connections and retrieving data. The Business Data Catalog bridges the gap between the portal site and your business applications and enables you to bring in key data from various business applications to SharePoint Server Web Parts, lists, search, user profiles, and custom applications.

The Business Data Catalog design is based on standardized metadata that describes the location and format of a back-end system and data entities defined within it. The Business Data Catalog also provides an execution component that is capable of reading metadata and that is able to retrieve external data from back-end systems and return that data to SharePoint Server in a standardized format.

Figure 4 shows the high-level architecture. As you can see, connectivity with traditional line-of-business systems is achieved by using standard Web services. Connectivity with database systems is achieved by using ADO.NET providers.

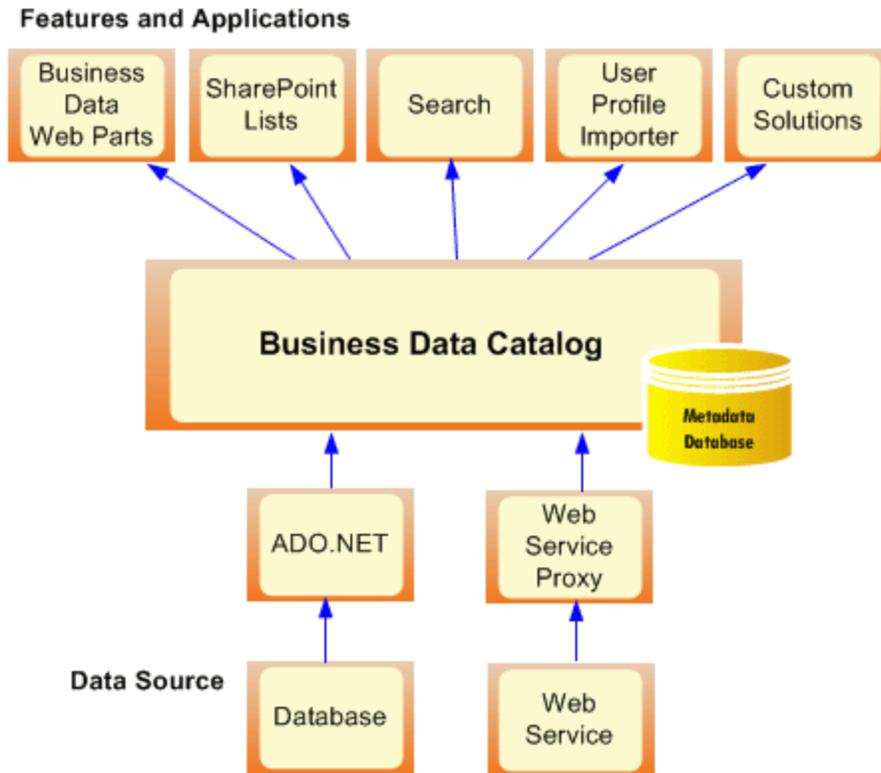


Figure 4: High level architecture of the Business Data Catalog

### **Metadata File Overview**

The first step is to describe the information being supplied by the back-end system, and to do so in a way that will be meaningful to business users. Back-end systems are usually exposed via a set of data tables and/or Web services that are focused on tasks. The need here is to describe them in terms of *entities*, things which business users will want to browse, search, or on which they might want to take action (examples would include Customer, Order, Account, etc.). This certainly includes an entire object's properties (fields), but it can also include associates between entities, actions that can be performed on the entity, and much more. These descriptions are provided in the form of XML files known as *Application Definitions*.

### **Methods, Actions, and Other Business Data Catalog Elements**

The definition of an entity contains identifiers, properties, and methods. The methods define how the Business Data Catalog interacts with entry points exposed by the back-end system. For a back-end system accessible through Web services, methods define the names of the Web service operations and the parameters required to call them. For a back-end system that is a database such as SQL Server or Oracle, methods define the names of stored procedure and SQL statements.

Entities can also define actions. An action is used to dynamically parse together the URL behind a hyperlink that allows a user to navigate from a page in a SharePoint Server 2007 portal site to another location. For example, an action defined on a Business Data Catalog customer entity could be written to

redirect users to a Web page in an SAP application that supports updates to customer information. Actions were designed to support scenarios where the Business Data Catalog is used to display read-only data and to link the user to another page, perhaps even another application, when updating or some other type of external operation is required.

### **Using Business Data inside SharePoint Server**

Once you have the XML file with the required metadata for a back-end system, you must import it into the BDC. BDCs are Shared Services, which means that they can be shared between multiple portals and managed without needing full portal administration privileges. Within the scope of a particular SharePoint Shared Service Provider (the management console for a set of shared services), XML application definitions are imported to create what is known as a Business Data Catalog application.

Once you have imported the required metadata, there are five out-of-the-box techniques to display the Business Data Catalog application's data within a portal site (these techniques are shown at the top of figure 4 above). SharePoint Server 2007 ships with a set of Business Web Parts that can be quickly added to pages to query and display Business Data Catalog data. Figure 5 below show an example of a dashboard page that uses connected Business Data Catalog Web Parts to dynamically interact with actual SAP information. In this case, the Business Data Catalog takes care of the communications and connection management with no code, and the Web Parts and dashboard page provide a ready-made user interface for this information.

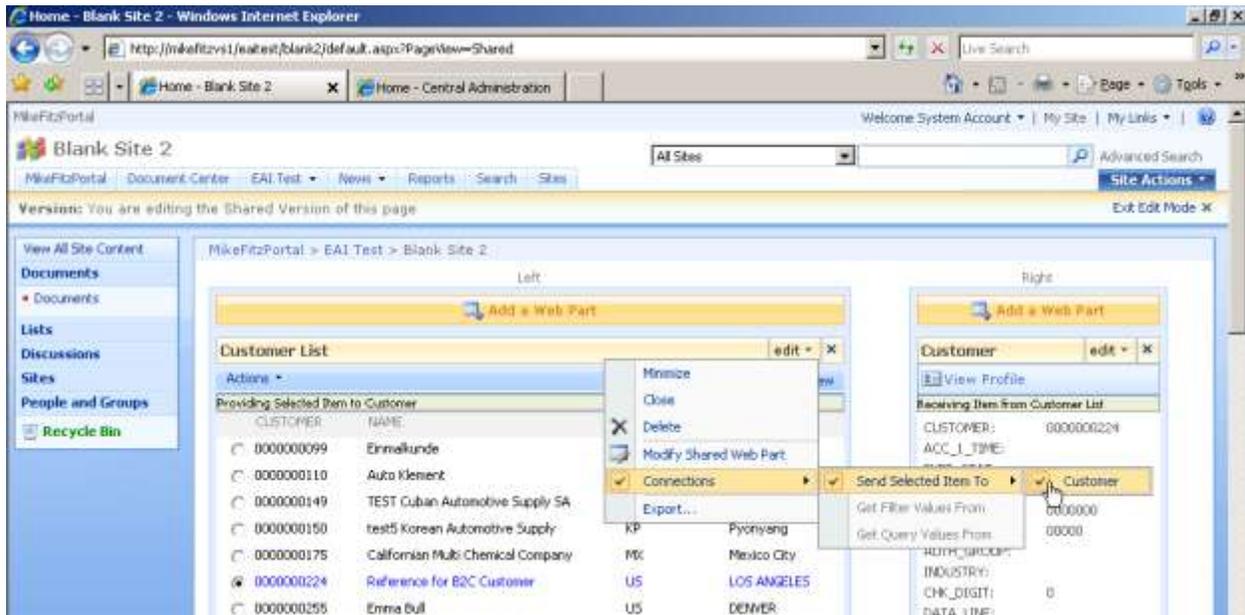


Figure 5: BDC Web Parts connected to each other in a master-detail relationship

You can also add new columns to lists and document libraries based on an entity defined in a Business Data Catalog application. A user editing a column based on an entity is automatically presented with a user interface making it possible to query the back-end system, as shown in figure 6.

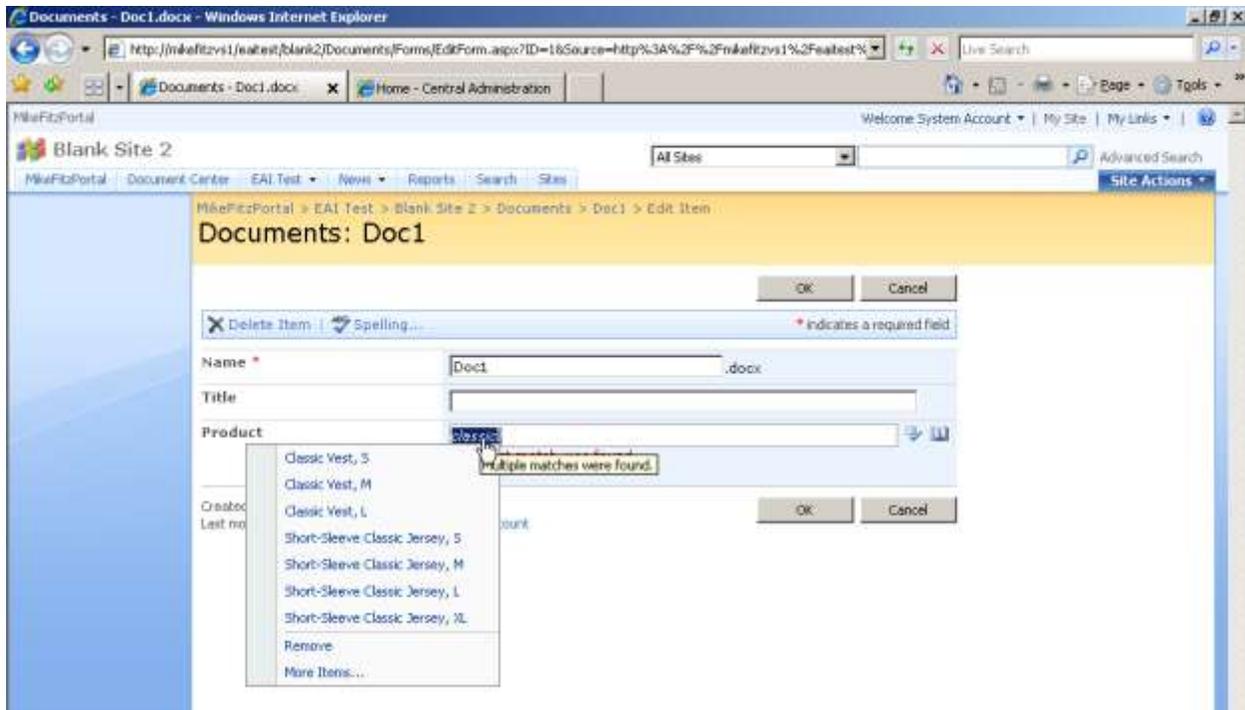


Figure 6: Name resolution in a SharePoint list item against the Business Data Catalog

As described in the workflow section below, list and library items with SAP information can take full advantage of built-in support for workflows.

The Business Data Catalog provides convenient features to map data from an entity to properties in a user profile and to synchronize this data at periodic intervals. For example, SAP employee data such as phone numbers or social security numbers can be included directly in user profiles, appearing alongside other information such as what is imported from Active Directory.

Business Data Catalog entities can also be accessed programmatically using custom code written against its object model. This makes it possible to write custom Web Parts as well as other server-side components and services that execute their own queries. One nice aspect of writing code to query Business Data Catalog entities is that you don't have to worry about managing connections or whether you are accessing the back-end system through Web services or ADO.NET. All those details are abstracted away by metadata and the Business Data Catalog execution engine.

Finally, the Business Data Catalog integrates with the SharePoint Server 2007 Enterprise Search service. SAP can be defined as a content source so that the SharePoint Server indexing service will crawl through its data and build indexes for the search engine. You do not need to write custom protocol handlers or IFilters or create searchable HTML representations of information in a database. This becomes a powerful feature because it allows users to discover data about things like customers and invoices when running standard search queries.

By making it possible to index line of business data from multiple systems, and by taking advantage of new features that provide a consistent and familiar search experience, increased relevance of search results, the ability to map custom search terms to content source properties, and new functions to search for people and expertise, and improved manageability and extensibility, Enterprise Search helps to do away with the traditional “information silo” situation most customers experience.

Another important feature of Enterprise Search, especially in the context of making SAP data searchable, is custom security trimming, which allows customers to trim search results at query time based on custom authentication types.

## **Workflow**

With the integration of Windows Workflow Foundation into SharePoint Server 2007, SAP information brought into the portal via the Business Data Catalog can now become part of powerful information worker-centric business processes. Furthermore, by using either custom “out of the box” workflows, or by using SharePoint Designer to create code-free custom workflows, power users can start defining their organizational level workflows.

With extended support for the Windows Workflow Foundation development model in Microsoft Visual Studio®, enterprise level developers can design organization-wide workflows that are surfaced at the highest level of Office SharePoint Server. These workflows will contextually drive an organization’s business process at any level.

## **ASP .NET 2.0 Web Part control set**

ASP .NET 2.0 and its new Web Part control set are what drive the simplicity of these Web Parts. They make it easier to create rich portal applications that support customization and personalization. Probably the most significant feature of this infrastructure is its plug-ability. Instead of being tied to a particular serialization implementation and data store, the provider architecture makes it relatively easy to write the personalization data to any back-end data store that makes sense for your site.

## **Forms Services**

Microsoft InfoPath® 2003 introduced a powerful, rich client way to integrate line of business systems such as SAP, providing both read and write capabilities through Web services support. A new feature of SharePoint Server 2007 is InfoPath® Forms Services, a scalable, security enhanced, standards-based form solution that organizations can use to extend the reach of a forms-driven business process to anyone through a Web browser. Forms can be designed in the traditional InfoPath rich client or in the new Microsoft Visual Studio editor for complete control over form functionality as well as look and feel.

SAP customers can easily build InfoPath forms that consume and present Web services in a way that makes it easy for information workers to view and to make changes to SAP information. With Forms Services, information workers can use a 100% browser-based front-end to work with SAP information. For more information on how to build InfoPath forms that connect to SAP, see the white paper *Consuming SAP Enterprise Services in a Microsoft Office InfoPath Form*

<https://www.sdn.sap.com/irj/sdn/go/portal/prtroot/docs/library/uuid/e0421f9a-4c9e-2910-1085-a45f8e813324>.

## **Excel Services**

Like InfoPath Forms Services, which lets anyone with a browser use a published InfoPath form, Excel Services makes it possible to render models and presentation items from a user-created spreadsheet on a SharePoint Server, making it accessible to clients and applications either through a Web Part (Excel Web Access) hosted in a SharePoint site or through SOAP query (Excel Web Services). This enables users to define and host complex calculations for an application within an Excel spreadsheet.

In the context of SAP interoperability, this is a very powerful capability. With no code, users can consume, analyze, and model SAP information in Excel, displaying information in meaningful ways with charts, PivotTables, and the like, and then publish that workbook (or just a part of it, such as the chart only) to a SharePoint Server site. Others can then view and interact with that information directly in the browser, where they see it in relation to other relevant information (for example, a Customer Sales chart shown on a portal page about a customer, where they can also see key customer contacts from SAP rendered in a Web Part, recent customer satisfaction data rendered in another Web Part, and so on).

## **Single Sign On**

Single Sign On is used to map the identity of a user who has logged on to a SharePoint Server 2007 portal site with a given set of credentials used by the user when logging into a back-end system. For example, imagine a user named Bob has two user accounts. First, he has a Windows account in Active Directory that he uses to log on to the local network and authenticate against the SharePoint Server 2007 Web server. Bob also has a second user account with different credentials that needs to be used when accessing SAP.

SSO solves the problem of allowing server-side code running on behalf of a user to seamlessly access the SAP system using SAP credentials once he has logged on to the portal site using his primary Active Directory account. SSO accomplishes this by providing a credential-mapping database to store his SAP user name and password in an encrypted format. Figure 7 below shows the architecture of the SSO service.

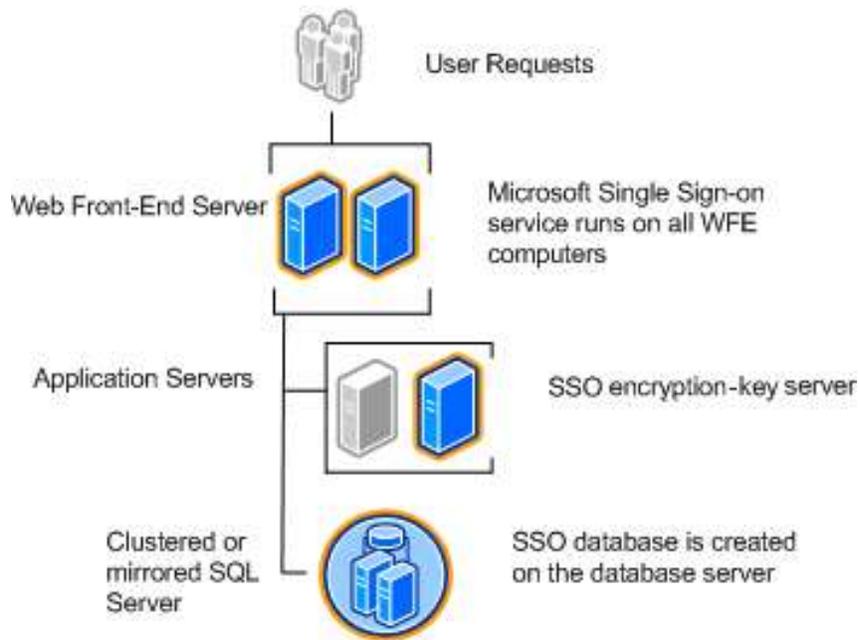


Figure 7: Single Sign-On service in an Office SharePoint Server 2007 server farm

Note that this credential mapping works for Active Directory groups as well, so members of a group will experience the same seamless access. SSO also provides the means for custom Web Parts and other SharePoint Server 2007 services such as the BDC, Excel Services, and Forms Services to retrieve user credentials that are required when accessing various back-end systems.

## Integration Options and Best Practices

Having looked at some of the possibilities with the Business Data Catalog, Enterprise Search, and other SharePoint Server capabilities, this section puts those possibilities within the context of what customers need to accomplish. It describes three basic methods of integration: surfacing SAP UI; integrating with SAP services; and surfacing SAP data directly without going through services. While describing these three levels of integration, we also describe how to best achieve them using combinations of SharePoint Server and other technologies and techniques.

The most basic method of integration involves page (or UI) redisplay - displaying SAP information inside custom ASP pages or Web Parts residing on SharePoint Server pages alongside other relevant information worker content.

The second method involves accessing SAP data using services and bringing it into SharePoint Server using some combination of the new platform capabilities. In this way, you can bring SAP information, and not just a view of it, into the context of the SharePoint Server experience. As we saw in the description of the Business Data Catalog, for example, it is possible to make SAP information a *dynamic* part of the metadata of documents in a SharePoint library, part of the data in a SharePoint list, part of what shows up in search results, and part of what drives and filters the content of other Web Parts.

Finally, in the third method, rather than surfacing views of SAP information or surfacing transactional data through services, you can surface just the data itself for the purpose of performing analysis. This usually involves exporting SAP data into a warehouse and then consuming the warehouse using Excel PivotCharts, SharePoint Server's business intelligence capabilities, and even the Business Data Catalog.

## Page Redisplay

There are a number of situations where simply displaying a view of SAP information inside a SharePoint Server page is sufficient integration. For example, customers may want to show a view of pending SAP expense reports using on their employees' SharePoint Server-based MySites.

### **IFrame Redisplay**

One way of surfacing SAP UI inside SharePoint Server is to use the new SAP iView Web Part, which can be easily configured to host an iView within an IFRAME. This approach is very useful in situations where you only need to display discrete iViews (that is, when you don't need to show multiple iViews and preserve the logical connections among them that the SAP portal provides). Figure 8 shows an example of surfacing one iView out of a mySAP page into a SharePoint page.

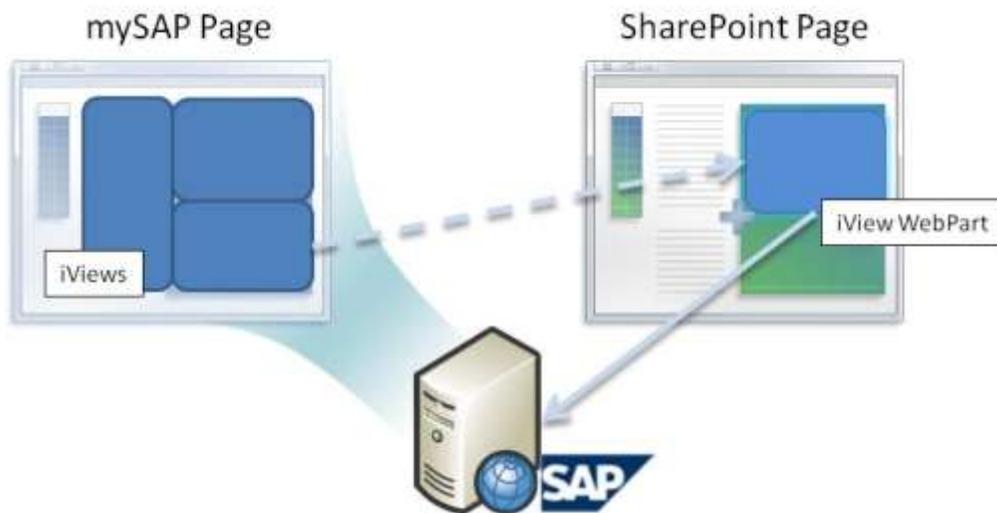


Figure 8: iView Web Part display in SharePoint

### **Web Services for Remote Portlets**

While the iView Web Part works very well for simple iViews, there are limitations even here, including the fact that you may have different UI controls and look and feel in the source and destination sites. Another, potentially more flexible method of including iViews would be to use Web Services for Remote Portlets (WSRP), a standard for embedding remote portlets.

WSRP defines a way that an Office SharePoint Server Web Part can be generated from a remote system. WSRP is preferable to the URL wrapping approach since it is a true Web service and not simply another system's HTML. However, there are caveats to implementing the WSRP approach with SAP.

- Passing identity information across systems in a seamless way can be a challenge.
- Only more recent versions of SAP support WSRP. For specific information about your version of SAP, consult your product documentation.

### **Business Server Page (BSP) Redisplay**

A strong alternative to redisplaying iViews and WSRP involves capturing and displaying entire SAP pages within Web Parts. SAP's classic page development environment is known as BSPs, or Business Server Pages. Business Server Page (BSP) applications are complete functional applications similar to the classic SAP R/3 transaction. Instead of exposing content using the SAP GUI client, the application is rendered in a Web browser. Figure 9 shows an example of surfacing a BSP in a SharePoint page.

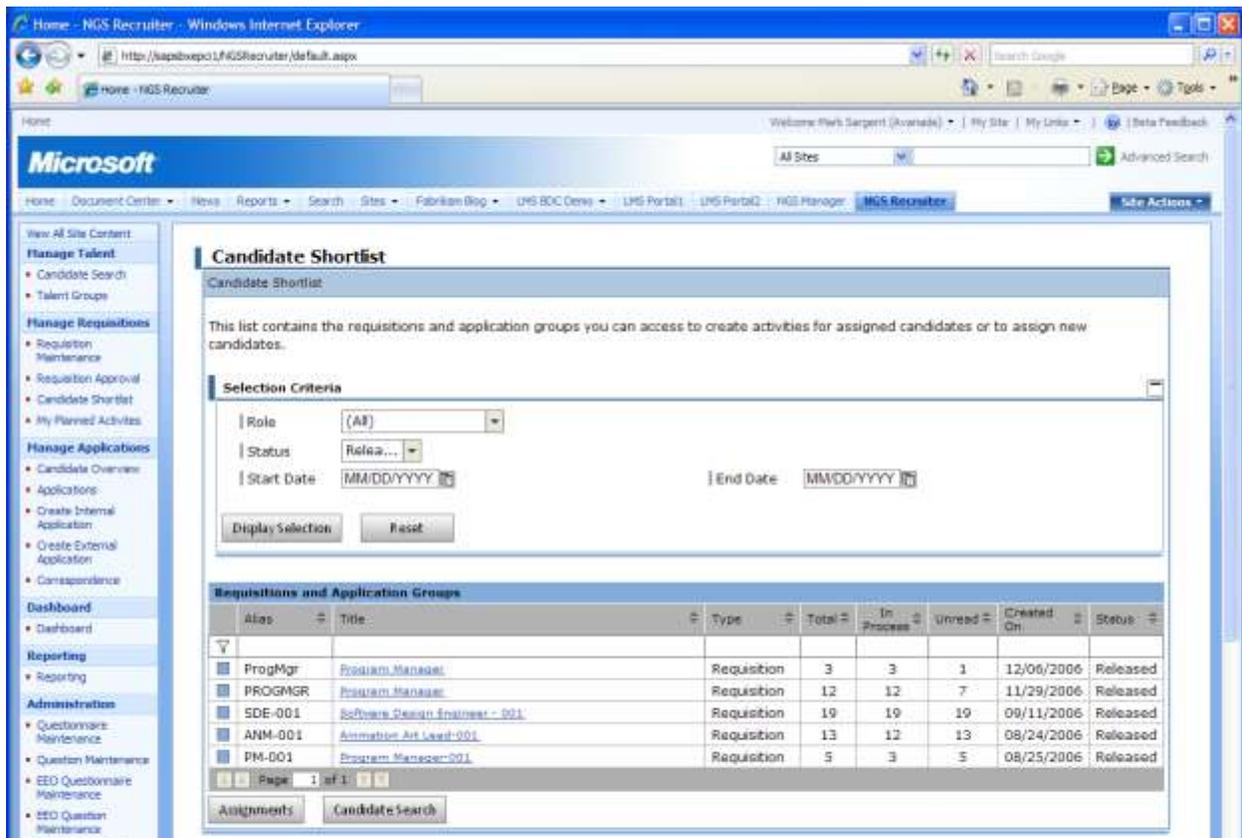


Figure 9: SAP E-Recruiting BSPs for Recruiters surfaced through SharePoint with SharePoint navigation

HTTP or HTTPS is used to access the application across the network, which means that standard products like firewalls and proxy servers can be used. The SAP Portal provides value through Single Sign-On and Session State management but is not required to execute BSPs. BSP's also support Java Server Pages. Using Active Server Pages, you can effectively wrap a BSP or JSP, thus maintaining the context of all the data in the source page and maintaining the connections among various page components as well. For more information on a real-world implementation of BSP redisplay, refer to

[Integration of SAP Business Server Pages \(SAP BSP\) in Share Point 2007](#), a paper about Microsoft's own experience and success.

## **Rich Collaborative Solutions through Integration with SAP Services**

More complex integration scenarios rely upon page redisplay as well, but they also require different approaches. In the business process example described earlier, SAP information needs to be incorporated into the unstructured parts of a business process for giving a sales quote to a customer. During the business process, critical activities such as searching for related information and people, doing analysis of the information, and then creating a quote document, all need a transparent integration of SAP information into the portal.

There are a number of methods for achieving parts of this vision, including the Portal Development Kit for .NET, which enables functional integration of SAP into Web Parts and portlets that can be used in both SharePoint Server and NetWeaver Portal.

In this example, however, we want to offer a vision for a more comprehensive integration strategy, one that brings SAP information directly into the context of SharePoint Server Web Parts, custom lists, dashboard pages, etc. The first goal for achieving these richer collaborative solutions is to work directly with SAP's LOB technology by integrating with SAP's services.

### **Pre-Composing Business Entities for Easier Consumption**

SAP's APIs are designed in a granular fashion to allow for maximum flexibility for application developers. As such, there is a general expectation that multiple APIs will need to be invoked as a set in order to accomplish a single unit of business functionality. With this in mind, attempting to bring SAP data into SharePoint Server by working directly against SAP's RFCs and BAPIs is not recommended. Note that, for future usage, SAP has just released the Enterprise Services Explorer, which integrates into Microsoft Visual Studio 2005, enabling developers to browse – from within the Microsoft environment – the SAP enterprise services in SAP's UDDI-based Enterprise Services Registry, and incorporate these into .NET applications, automatically generating proxies to those services.

Independently of Enterprise Services Explorer, there are five methods for pre-composing user-targeted business entities out of SAP's discrete APIs and exposing these resultant business entities as Web Services for usage by SharePoint Server's Business Data Catalog, InfoPath Forms Services, or other facilities. They are:

- **Custom BAPI objects** – it is possible to create a custom BAPI object that combines more discrete BAPIs into consumable business entities
- **Using SAP Enterprise Services** – available soon, but will only work with the latest deployments
- **Custom Web services** – this may be a more difficult approach, but, as part of a larger SOA effort, may also be the most comprehensive and flexible

- **Custom Microsoft BizTalk® Server orchestrations** – for environments in which BizTalk is already in use, creating dedicated orchestrations that manage the necessary RFC and BAPI calls is a very approachable solution and one that is easy to consume within SharePoint Server
- **Using 3<sup>rd</sup> Party products**

### **Consuming Pre-Composed SAP Business Entities in SharePoint Server**

With any of the 5 methods of pre-composition, the result is a middle tier that acts as a service layer to be consumed by front-end applications. One approach to consuming SAP information through such a middle tier is to create custom SharePoint Server-based user interfaces. This usually entails writing custom Web Parts that consume SAP information and make it possible to write back to SAP as well.

While the custom Web Part approach allows for highly customizable interfaces, using the Business Data Catalog can enable you to create highly functional user interfaces for the same information with little to no custom code. From there, you can start to construct a complete collaboration solution using the built-in SharePoint Server capabilities described earlier.

For example, one of the first things you can do is to build a custom Enterprise Search Content Source that crawls the SAP information. Defining the content source, any scoping we require, and any security around who gets to see what, is all done through the SharePoint Server UI. No coding is required. Of course, with pervasive XML support in SharePoint Server, the search results display can be associated with an XML Transformation file to provide a customized look and feel.

Using the Business Data Catalog means that the SAP information is now available to be used in Web Parts, in document metadata, and in other ways that help information workers analyze and process information. For example, a search for a customer name will return results in SharePoint Server search, and clicking on the customer name may open a so-called Profile Page, which displays dynamic information about that customer, such as recent orders, satisfaction ratings, and so on. This can all be displayed in out-of-the-box SharePoint Server Web Parts, such as Business Data Catalog Web Parts and Excel Web Access Web Parts, which can be configured with parameterized settings such that they all receive the same input parameter at runtime (the customer name, in this example). In sum, for research and analysis, SharePoint Server provides tools for deeply integrating SAP information with no code.

One area in which SharePoint Server really shines is document collaboration, and some new innovations in this space make integration using the Business Data Catalog extremely valuable. SharePoint Server 2007 introduces the concept of the Content Type, which could be a document template in a library that has associated with it a specific SharePoint Server-based workflow, some auditing policy, and some specific metadata properties. Anytime a user goes to create a new document based on that content type, all the settings defined by IT are enforced. When SAP information is available through the Business Data Catalog, it can be defined as part of the metadata for a content type.

### **Writing Back to SAP**

As valuable as bringing SAP information into MOSS through search, into Web parts, and into document metadata can be, sometimes it is necessary to make changes to that data and write those changes back

to SAP itself. While the BDC does not provide this capability, there are no- and low-code ways of doing this with Microsoft technologies. The easiest way is by creating a Microsoft Office InfoPath form, which can be hosted using Forms Services for a completely browser-based solution. The form can use existing SOA Web services to write information back to SAP. Again, for more information on how to build InfoPath forms that connect to SAP, see the white paper *Consuming SAP Enterprise Services in a Microsoft Office InfoPath Form* <https://www.sdn.sap.com/irj/sdn/go/portal/prtroot/docs/library/uuid/e0421f9a-4c9e-2910-1085-a45f8e813324>.

The net result is this scenario: a user creates a new quote document from a SharePoint Server library, and it not only loads the proper Microsoft Office Word template, but it also loads an information panel, in which the user provides metadata about the quote, such as Customer Name. When the user types the customer name, the string is checked against the Business Data Catalog and validated, eliminating typing errors. Further, the information panel can be configured to bring in additional information about the customer, such as annual sales to that customer, credit rating, and so on. This information automatically becomes part of the metadata of the document itself, and, because of the open XML formats of the 2007 Office System, this information can also be bound directly INTO the document, thus making it easier to create the quote, and eliminating errors.

When the user submits the document, SharePoint Server can execute a workflow, apply security and auditing policy and so on. Finally, as one of the final tasks of the workflow, another user clicks a link that opens a browser-based InfoPath form that is pre-populated with relevant project data by the workflow. The user completes the form and submits it, and InfoPath writes the changes back to SAP.

In other words, SharePoint Server enables the complete document lifecycle, and when users have SAP information integrated into SharePoint Server, it makes the business process more powerful and more unified.

## **Business Intelligence Using SAP Data**

One of the most powerful tools for the business decision maker is Microsoft Excel. With the ability to consume data directly from SQL Server Analysis Services (AS), Excel has remained the most common end-user BI tool. As a result, a great interoperability scenario involves exporting SAP transactional information into a SQL Server Analysis Services warehouse and then consuming it in Excel. This is a very simple approach, especially when write-back to SAP is not a requirement.

Once available in the SQL Server AS warehouse, some great new BI features of the Excel 2007 client program, such as Conditional Formatting and new PivotTable capabilities, can be used to visualize and analyze SAP information. Then, using Excel Services, workbooks can be published to SharePoint Server pages and can become part of dynamic dashboards (see figure 10).

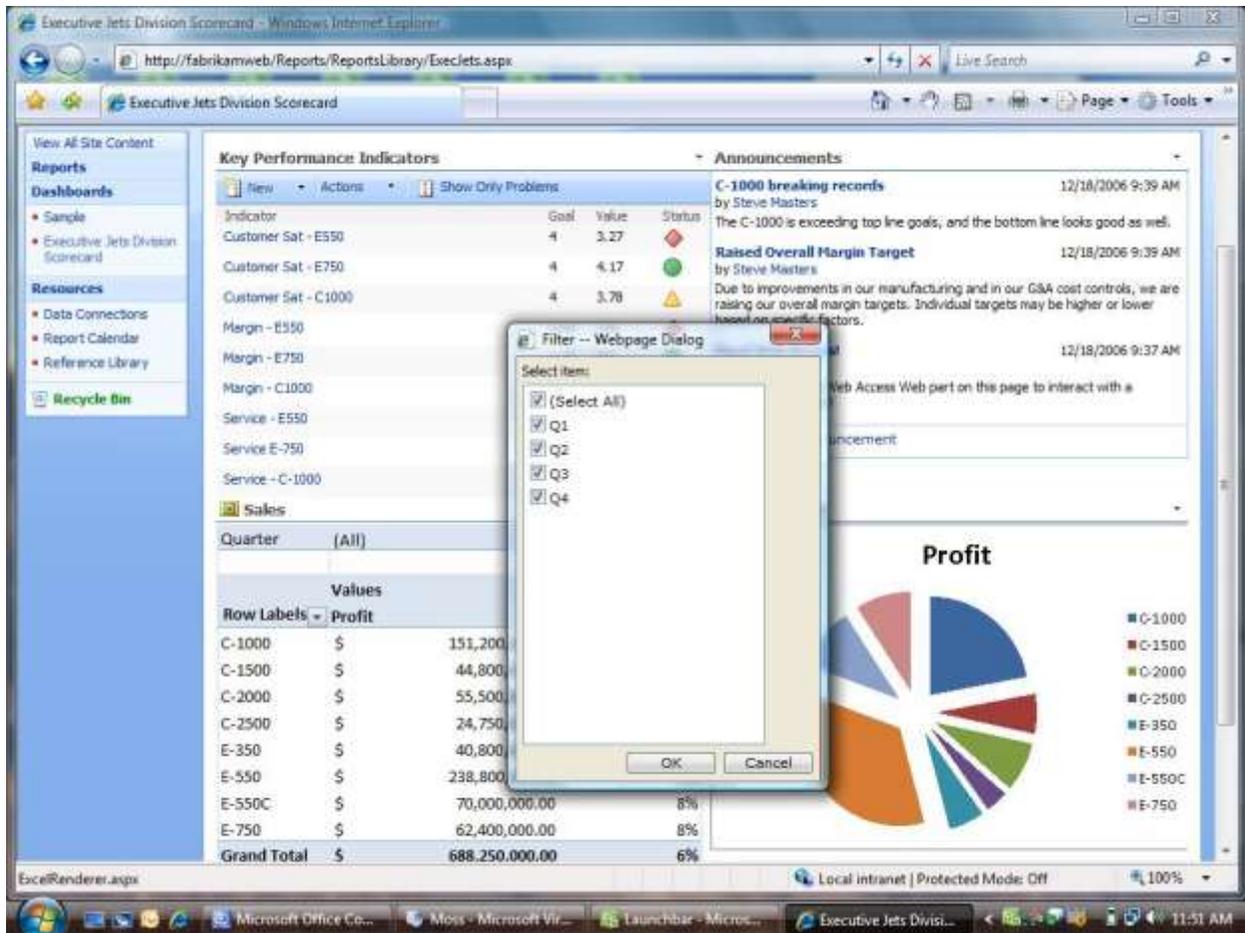


Figure 10: A SharePoint Dashboard with KPIs and Excel Web Parts using LOB data

Note the filter dialog in the figure. Excel Web Parts let you interact with published workbook, enabling Web-based analysis.

Other powerful dashboard capabilities can also consume SAP information in SQL Server AS. For example, as shown in figure 10, new Key Performance Indicator (KPI) capabilities in SharePoint Server can pull trend information from SQL Server AS and present graphical indications of performance right on a dashboard page.

Note that SharePoint can also directly connect to SAP Business Warehouse data using the .NET Data provider for NetWeaver BI which is Web Services based.

## Conclusion

SAP customers appreciate the tremendous value of their SAP systems, but they also recognize that they can get more out of them by bringing SAP data into the everyday activities of information workers. Microsoft Office SharePoint Server 2007 provides a rich infrastructure for integration of the two worlds in which information workers live: the worlds of structured and unstructured activities.

This paper has described what is possible at a high level. For more information about how to actually achieve the benefits of integration, use the resources in the Call to Action section.

## Call to Action

To learn more about developing solutions that bridge the gap between the structured world of SAP and the unstructured world of information work, explore the following resources:

- Paper: *Enabling “Real World SOA” through the Microsoft Platform* - <http://download.microsoft.com/download/b/4/d/b4db580a-0361-4907-9a6e-9d2866d8b581/Real%20World%20SOA.doc>
- SharePoint Server 2007 SDK - <http://msdn2.microsoft.com/en-us/library/bb226704.aspx>
- Paper: *Building Office Business Applications: A new breed of business applications built on the 2007 Microsoft Office system* - <http://go.microsoft.com/?linkid=5078276>
- Business Productivity Infrastructure Optimization: Self-Assessment - <http://labs.microsoft.com/business/peopleready/bizinfra/ac/bpio.mspix>
- Microsoft BizTalk Server: Server Oriented Architecture - <http://www.microsoft.com/biztalk/solutions/soa/default.mspix>
- The Microsoft-SAP Customer Information Center <http://www.microsoft.com/SAP>
- Duet for Microsoft Office and SAP <http://www.duet.com/>
- Paper: *Integration of SAP Business Server Pages (SAP BSP) in Share Point 2007* - [http://download.microsoft.com/download/4/3/d/43de0f43-df6a-4cf2-8c91-2c9001df5945/CB\\_027\\_Integration\\_of\\_SAP\\_BSP\\_into\\_SharePoint\\_Server\\_2007.pdf](http://download.microsoft.com/download/4/3/d/43de0f43-df6a-4cf2-8c91-2c9001df5945/CB_027_Integration_of_SAP_BSP_into_SharePoint_Server_2007.pdf)
- Paper: *Building Custom Search WebParts with Integrated SAP NetWeaver Portal Search for Microsoft Office SharePoint Server 2007 (MOSS)* - [http://download.microsoft.com/download/5/7/f/57f1490e-8a8d-497b-bbae-ec2a44b3799f/MOSSEnterpriseSearch\\_NetWeaverPortal.pdf](http://download.microsoft.com/download/5/7/f/57f1490e-8a8d-497b-bbae-ec2a44b3799f/MOSSEnterpriseSearch_NetWeaverPortal.pdf)

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

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

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

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

© 2006 Microsoft Corporation. All rights reserved.