

# Interoperability between SAP Enterprise Portal 6.0 and Windows SharePoint Services

Version 1.0, September 2005

Thomas Meigen, Project Director CTSC, SAP AG  
Iwan Kostin, Consultant NetWeaver/Enterprise Portal, SAP Germany  
Michael Lechtenböcker, Senior-Consultant – CEO, Lechtenböcker, Schmidt und Partner

## Summary

The document addresses several questions in the context of SAP Enterprise Portal and Windows SharePoint Services co-existence. It covers the three main areas Document Management Interoperability, Front-end and Content Integration and Collaboration Interoperability. While the focus is on providing general guidance about integration opportunities, the document in some cases also offers technical How-To's.

## Applies to

- SAP NetWeaver 04
- SAP Web Application Server 6.40, SP 9
- SAP Enterprise Portal 6.0, SP 9
- SAP Knowledge Management & Collaboration Component 6.0, SP 9
- Microsoft Windows SharePoint Services, SP 1

## Keywords

SAP NetWeaver, Enterprise Portal, Windows SharePoint Services, WebDAV, WSRP, Portlet, Collaboration

## Level of difficulty

IT Management, Technical Architects, Technical consultants, Developers

Copyright 2005 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, and Informix are trademarks or registered trademarks of IBM Corporation in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

MaxDB is a trademark of MySQL AB, Sweden.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

# Contents

<b>Summary</b> .....	<b>1</b>
<b>Applies to</b> .....	<b>1</b>
<b>Keywords</b> .....	<b>1</b>
<b>Level of difficulty</b> .....	<b>1</b>
<b>Contents</b> .....	<b>3</b>
<b>Introduction</b> .....	<b>4</b>
<b>Document Management Interoperability</b> .....	<b>7</b>
<b>The SharePoint WebDAV Connector for SAP Enterprise Portal</b> .....	<b>8</b>
<b>Features of integrated SharePoint document libraries</b> .....	<b>8</b>
<b>Usability Considerations</b> .....	<b>10</b>
<b>Multiple locations and multiple sites</b> .....	<b>13</b>
<b>Embedding of the SharePoint user interface into SAP KM</b> .....	<b>13</b>
<b>Front-end and Content Integration</b> .....	<b>15</b>
<b>Portal Component Development</b> .....	<b>15</b>
Portal Development Kit for Microsoft .NET (PDK for .NET) .....	16
Web Part Development using Visual Studio .....	18
Web Services based integration .....	19
Web Service Support by SAP Enterprise Portal .....	20
Web Service Support by Windows SharePoint Services .....	21
<b>Portal Component Syndication</b> .....	<b>22</b>
Web Services for Remote Portlets (WSRP).....	22
<b>Portal Component Embedding</b> .....	<b>23</b>
Embedding customized Web Part pages or lists into SAP Enterprise Portal .....	23
Capturing SharePoint content into SAP Enterprise Portal .....	25
SAP iView Web Part Toolkit for SharePoint Products and Technologies .....	26
The toolkit automates the process of embedding the iView into WSS. It does not fix the general issues of portal component embedding. ....	26
<b>Portal Embedding</b> .....	<b>26</b>
<b>SharePoint sites embedded in SAP Enterprise Portal</b> .....	<b>27</b>
<b>Portal Launch</b> .....	<b>27</b>
<b>Collaboration Interoperability</b> .....	<b>28</b>
<b>How-To</b> .....	<b>29</b>
<b>How-To: Simple URL Integration</b> .....	<b>29</b>
SharePoint sites integrated in SAP Enterprise Portal .....	29
Integration of a SharePoint List View into the SAP Enterprise Portal .....	31
Capturing SharePoint Content into Web Parts integrated in SAP Enterprise Portal .....	35
<b>How-To: Configuring the SAP WebDAV Repository Manager to integrate SharePoint..</b> <b>41</b>	<b>41</b>
Define an HTTP system in the CM system landscape .....	41
Create a cache for the WebDAV repository manager.....	42
Configure the WebDAV repository manager.....	42
Create a WebDAV system template .....	43
Create a WebDAV system in the portal system landscape .....	46
Specify user mapping.....	47
<b>How-To: Security</b> .....	<b>49</b>
SSO22KerbMap .....	49
SSO for internal service users .....	50
Microsoft Enterprise Single Sign-On Service provided by SharePoint Portal Server .....	51
<b>Conclusion</b> .....	<b>52</b>
<b>References</b> .....	<b>52</b>

## Introduction

Recently some joint announcements by SAP and Microsoft created a lot of attention as well as the expectation that all products from these two companies should work seamlessly together. While this is true in many cases there are repeatedly questions in our customer base about how to approach coexistence specifically between SAP Enterprise Portal and Microsoft SharePoint Products and Technologies.

- **SAP introduced SAP Enterprise Portal (SAP EP)** as part of SAP NetWeaver as the strategic user interface for SAP users. SAP NetWeaver was presented in 2003 and since then more and more SAP customers roll out SAP Enterprise Portal to their employees offering them a centralized, role-based entry point into most of the applications for their working day.
- Also in 2003 Microsoft presented Office System 2003 including **Windows SharePoint Services (WSS)** and **SharePoint Portal Server (SPS)**. While Windows SharePoint Services, which are part of Windows Server 2003, primarily focus on team-level collaboration based on tight integration with Microsoft Office, SharePoint Portal Server integrates many of these Windows SharePoint Sites and simplifies access to them through search and navigation capabilities. Being part of Windows Server 2003 WSS do not have to be licensed separately.

Even though both products at a first glance seem to serve a similar purpose both have been designed and implemented with different goals in mind and using different technologies. While SAP Enterprise Portal (coming from the worlds largest provider of business software solutions) clearly focuses on providing an integrated access to SAP and non-SAP business applications, SharePoint Products and Technologies (coming from the worlds largest provider of operating systems and productivity tools) emphasizes the collaboration around documents and unstructured data. Common to both products is that they also represent portal platforms that can be extended through configuration and through development based on open standards and protocols.

A seamless integration with the respective partner's product was not a primary design objective. Some integration opportunities will be addressed out of the box (example: WebDAV Connector for Windows SharePoint Services), others may not.

Still many people in companies and organizations want to better understand which technical links exist between these products and how to make best use of this interoperability for their specific needs. This document will try to answer some of these questions and offer guidance for co-existence scenarios between Enterprise Portal and SharePoint. Unless explicitly states otherwise, the paper addresses the integration with Windows Sharepoint Services, which we for simplification reasons will refer to as Sharepoint or WSS. Many of the findings in this document may also be applicable to Sharepoint Portal Server (SPS) since WSS and SPS build on the same foundation, but we validated them only for WSS.

We will address the 3 main interoperability areas:

### 1. **Document Management Interoperability**

Both EP and WSS offer document management capabilities. Here a possible coexistence of both products will be the main topic. In particular the use of the upcoming WebDAV Connector for Windows SharePoint Services will offer a direct link and therefore will be investigated.

## 2. **Front-end and Content Integration**

"Can we use SAP business processes in SharePoint?" or "How can we bring Microsoft content into Enterprise Portal?" are frequently asked questions. We will describe what is possible and useful but also what perhaps is not.

## 3. **Collaboration Interoperability**

Again, both products offer similar capabilities. Are these interoperable?

At the end of the document section **4. How-To** contains detailed step-by-step guides and background explanations for selected topics.

While not all possible topics may be contained in this document, we still believe that some key questions are being addressed. This does not mean that there always is a solution for all desirable interoperability scenarios. But sometimes a clarifying statement like "No, this won't work." also can be helpful.

Let's start with a brief explanation what SAP Enterprise Portal and SharePoint Products and Technologies are:

### **SAP Enterprise Portal**

SAP Enterprise Portal, which is part of SAP NetWeaver, provides a complete portal infrastructure and includes knowledge-management and collaboration functionality. It allows companies and organizations to quickly and effectively integrate SAP solutions, third-party applications, legacy systems, databases, unstructured documents, internal and external Web content, and collaboration tools.

SAP EP uses open standards, Web services, and tight integration with other SAP NetWeaver components to support heterogeneous systems from all major technology providers. Built in Java SAP EP offers comprehensive support for Java, Java 2 Platform Enterprise Edition (J2EE), and Microsoft .NET technology as well.

At the same time SAP EP already comes with predefined business content. Designed specifically to meet the needs of various roles, this content consists of thousands of iViews (SAP name for portal components) and is bundled into more than 70 multi-language business packages offering access to SAP and partner solutions. This content can be used without the need for additional development. *(Source: SAP website)*

### **Windows SharePoint Services**

Windows SharePoint Services allows teams to create Web sites for information sharing and document collaboration, benefits that help increase individual and team productivity. Windows SharePoint Services is a component of the Windows Server 2003 information worker infrastructure and provides team services and sites to Microsoft Office System and other desktop programs. It also serves as a platform for application development. Including such IT resources as portals, team workspaces, e-mail, presence awareness, and Web-based conferencing, Windows SharePoint Services enables users to locate distributed information quickly and efficiently, as well as connect to and work with others more productively. *(Source: Microsoft Windows Server website)*

## **SharePoint Portal Server**

SharePoint Portal Server 2003 capitalizes on Windows SharePoint Services by using SharePoint sites to create portal pages for people, information, and organizations. While these sites are specific to SharePoint Portal Server 2003, they use technologies such as Web Parts (which is Microsoft's name for portal components) and SharePoint document libraries that are delivered by the Windows SharePoint Services platform. This integration greatly reduces the time and cost associated with development, training, and support.

SharePoint Portal Server 2003 also extends the capabilities of Windows SharePoint Services by providing organization and management tools for SharePoint sites. In addition, SharePoint Portal Server 2003 enables users to publish information and documents stored in their Windows SharePoint Services sites to the entire organization. (*Source: Microsoft SharePoint website*)

## Document Management Interoperability

In today's information-driven economy, it is viable for a company to recognize, share and manage intellectual property generated by its employees. The process that lets companies benefit from its intellectual assets is defined as Knowledge Management. An effective and creative approach to knowledge management has a direct impact on a company's success because it can:

- support innovation by exchange of information and ideas
- streamline processes by providing required information at the right time

Both Microsoft and SAP as part of their solution stack offer functionality to address knowledge management and document management requirements.

### **SAP Knowledge Management**

The knowledge management capabilities of SAP NetWeaver – called SAP Knowledge Management (KM) - are delivered by SAP Enterprise Portal. SAP Knowledge Management provides capabilities for a role-specific access to unstructured information from various data sources in the portal. This is done by creating an additional layer on top of existing document repositories and connecting them via connectors – Repository Managers – to this layer.

For all these integrated data sources, Knowledge Management offers plenty of additional generic services for contained folders and documents comprising

- Document authoring and publishing
- Version management
- Powerful search and navigation through taxonomies
- Intelligent publishing, automatic classification and subscription
- Flexible and configurable user interface

The mentioned repository managers for connecting external data sources are either delivered by SAP (e.g. WebDAV, File System) or should be programmed by partners and/or customers (e.g. Lotus Domino, IXOS). Capabilities of a particular repository manager depend on its implementation or on the type of the corresponding subsystem.

### **Windows SharePoint Services**

Windows SharePoint Services takes a different approach by offering a platform for information sharing and document collaboration. Windows SharePoint Services provides mechanisms for storing and managing information and documents while at the same time offering deep integration with Microsoft Office. WSS puts more emphasis on collaboration than on document management. The main building block in WSS is a site. A SharePoint site is a container for lists, document libraries and other sub-sites. While lists allow users to collect and share information in a table-like form, document libraries provide storage for physically existing documents.

Regarding the targeted integration between SAP Knowledge Management and Windows SharePoint Services discussed in this chapter, a document library is the central entity. Within the scope of this document, it is not supposed to provide access to SharePoint's lists.

## The SharePoint WebDAV Connector for SAP Enterprise Portal

As part of the joint initiative around Web services for the Enterprise in 2004 it was also announced that Microsoft will offer the missing link between the knowledge management capabilities of SAP Enterprise Portal and Windows SharePoint Services. This link will be implemented as a SharePoint WebDAV Connector. The Connector will make it possible to access document libraries in Windows SharePoint Services from within KM. KM user will be able to add, modify and search documents and manipulate their properties from within KM iViews even if they are located in a Windows SharePoint Services site.

As the SharePoint WebDAV Connector currently is not yet available, it is not possible to document all details of its installation, configuration and usage. However, because it will be accessed from within SAP Enterprise Portal through the open standards protocol WebDAV, we can already now make fairly firm assumptions about its behavior.

SAP Knowledge Management can be used as a WebDAV server or a WebDAV client. In a scenario as a WebDAV client, SAP KM provides an access to documents stored on other WebDAV-enabled servers. For this purpose, you use the WebDAV repository manager.

The setting up the access to documents in SharePoint Document Libraries does not differ from the default procedure and comprises following steps:

- Define an HTTP system in the CM system landscape.
- Create a cache for the WebDAV repository manager.
- Configure the repository manager parameters for the WebDAV repository manager.
- *Optional: Create a WebDAV system template.*
- *Optional: Create a WebDAV system in the portal system landscape.*
- *Optional: Carry out user mapping for the WebDAV system.*

### **Note:**

The last three steps can be left out if the SSO22KerbMap module is used to implement ticket-based Single Sign-on. They would only be required in conjunction with user mapping.

## Features of integrated SharePoint document libraries

Once required SharePoint document libraries are integrated in SAP Knowledge Management through the WebDAV repository Manager, SAP EP users are able to apply KM features on documents contained in them.

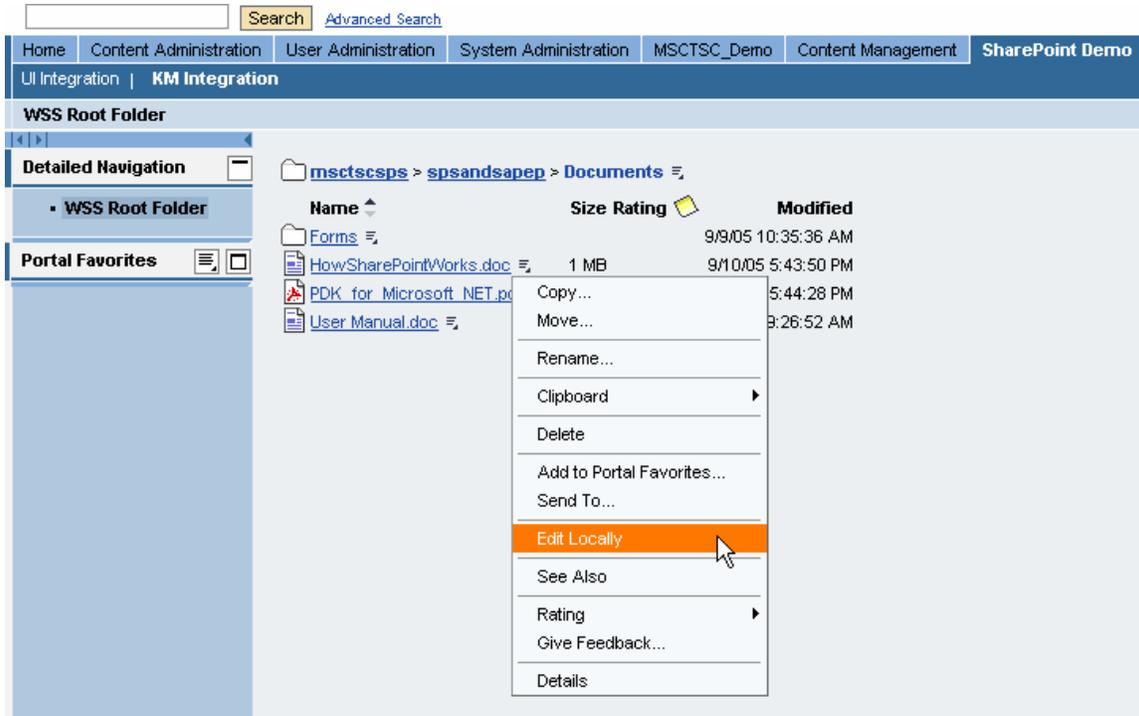
### **Note:**

The functions of the WebDAV repository manager depend on the capabilities of the WebDAV remote server.

By default, a SAP KM user can browse the structure of integrated SharePoint sites, view contained documents and even upload new documents. By using the upcoming SharePoint WebDAV Connector following features will also become available to SAP KM users:

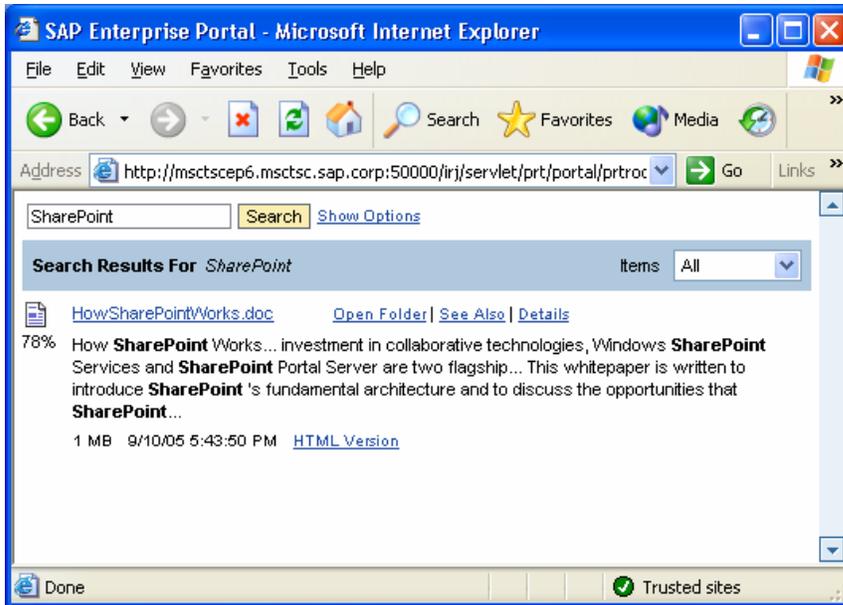
- Versioning
- Check in / out
- Viewing and changing metadata
- Copy / move of documents

The handling of all listed features is described in [SAP KM User Guide](#).



Picture 1: SharePoint based documents accessible through KM iView

Additionally, a system administrator will be able to create an index for integrated SharePoint document libraries, so that SAP EP will be able to search for documents contained there or to find them browsing through taxonomies.



Picture 2: SharePoint based document found by TRex search

How to create and manage indexes and taxonomies in SAP KM is described in the [KM Content Manager Guide](#).

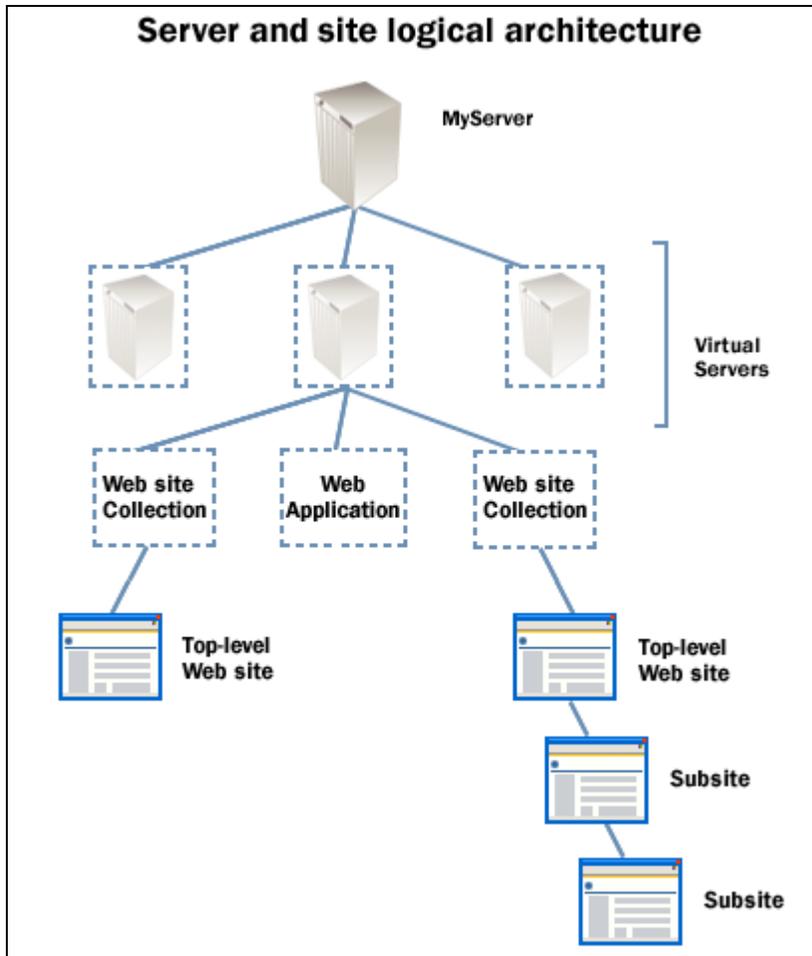
## Usability Considerations

When integrating SharePoint document libraries into SAP Knowledge Management it must be kept in mind that Windows SharePoint Services has the ability to host a very large number of sites. Sites can be well-defined sites for real or virtual teams or project. In addition sites can be created for various other purposes:

- Team Sites
- Document Workspace
- Meeting Workspace

In larger organizations there quickly can be thousands of SharePoint sites and the question arises how to structure access to them. For this purpose Microsoft offers SharePoint Portal Server. Another option could be SAP Enterprise Portal in conjunction with the WebDAV Connector for Windows SharePoint Services.

The following picture demonstrates the logical architecture of WSS on a single server.



Picture 3: Source: Windows SharePoint Services Administrator's Guide

Because for all nodes (Web site collection, Top-level Web or Subsite) in the hierarchy WSS offers dedicated URLs, each node can be registered in SAP KM individually as a repository. Such a repository will include all child nodes (subsites) as well, including all contained document libraries.

It is evident that it would not be sensible to make all these sites available to KM users through one repository. Organizational precautions are required to identify an appropriate subset of Windows SharePoint Sites for exposure through SAP KM. This means the KM administrator should only select branches of the site hierarchy that:

- limit the number of document libraries per repository
- contain relevant documents for EP users
- maintain the usability inside of KM iViews for the end-user

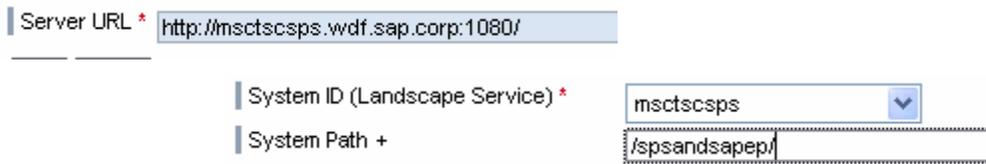
This approach is necessary both for practical and performance reasons.

**Note:**

You cannot address a specific document library but sites containing document libraries.

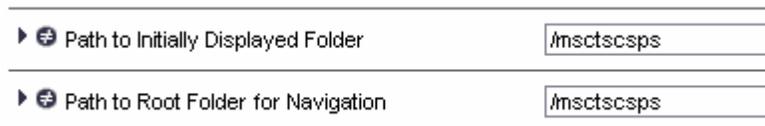
There are some parameters in the KM configuration and in KM Navigation iViews that are related to this topic:

- As outline above, configuring a WebDAV repository manager to access SharePoint's documents, you can either choose the root folder of a virtual server or specify site collection or subsite as entry point for a KM repository. In the configuration of the WebDAV Repository Manager you define the root folder of a virtual server as entry point by entering hostname and port number for the parameter *Server URL* in the HTTP System configuration (e.g. *http://myserver:8080*) and leave the parameter *System Path* unspecified. To select a child node in the site hierarchy the parameter *System Path* needs to be specified for a dedicated site collection or site (e.g. */yoursite*).



Picture 4

- Creating a KM Navigation iView, you can select which documents or folders are visible to the end user. You achieve it by specifying the iView parameters *Path to initially displayed folder* and *Path to root folder for navigation*. If there is a need for more sophisticated customization of the user interface, you can use the ability of KM's flexible user interface.



Picture 5

**Note:**

Limitation of document visibility for end users has no impact on access rights in KM. That means that users still can access documents even if they are not visible to them (e.g. through the document search).

## Multiple locations and multiple sites

SharePoint Portal Server 2003 provides the Shared Services infrastructure for supporting organizations with a divisional structure and multiple geographic locations. Shared Services installed on dedicated farms support specific types of SharePoint sites and benefit organizations that must:

- Address throughput and availability requirements for the portal site and team Web sites independently
- Enforce different security and customization standards for the portal site and the team Web sites
- Address the need for a large number of portal sites or centralized personal sites by using shared services

SharePoint Portal Server 2003 Shared Services are an important concept for medium and large sized organizations with multiple portal sites who want to reduce deployment, management, network and server costs.

The complexity increases when an organization operates WSS and SAP Enterprise Portal in parallel. This chapter should not try to discuss all challenges the organization would face in such environment, but rather we want to point out a possible solution for following scenario:

*Both portals – SAP Enterprise Portal and Microsoft SharePoint Portal Server – are hosted on different locations and are connected through Wide Area Networks. A user located on the same LAN segment as the SharePoint server needs to use documents from a SharePoint library, which are related to a business process in the SAP Enterprise Portal. The user is connected to SharePoint through a LAN connection on the one hand, and on the other hand he is connected to the Enterprise Portal by being routed through a WAN.*

By default, all document requests are handled by Knowledge Management causing unnecessary round-trips in the described scenario. In order to eliminate this performance relevant issue, users should request documents with its original URLs.

Changing the default behavior requires implementation of a property filter. In general, repository filters allow the manipulation of repository framework objects as they are passed through the repository framework. Property filters hide or modify existing properties of resources, or add virtual properties to resources.

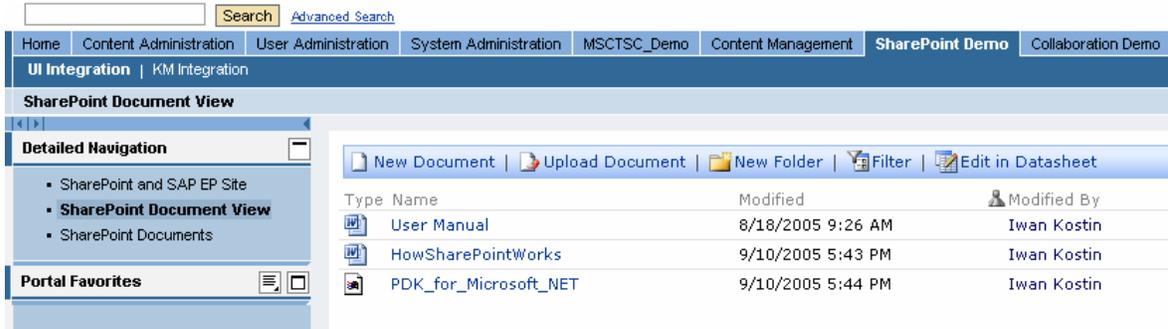
The property filter to implement for this scenario would be applied on every read access and would rewrite the document property `rnd:contentlink` with the original URL.

In order to experience more about the development for Knowledge Management and Collaboration please read the [corresponding guide on SDN](#).

## Embedding of the SharePoint user interface into SAP KM

Besides offering access to SharePoint document libraries through WebDAV it is also technically feasible to plug the document library Web Part into Enterprise Portal. Within an URL iView the SharePoint URL to an existing site, list or library is provided. Thus the SAP EP user is able to navigate through the SharePoint content from within SAP EP. How this kind of UI integration can be implemented and customized is described in the How-To section of this document.

However, this approach is inferior to WebDAV based integration. Displaying views on document libraries within URL iViews in SAP Enterprise Portal represents a loose type of integration. Though SAP EP users can access documents in SharePoint, they are not able to use KM services like search and classification. The documents are decoupled from the business process.



Picture 6: Document library Web Part embedded in Enterprise Portal

## Front-end and Content Integration

A very obvious question is the one how web content from EP and WSS can be leveraged in the peer product. This is a kind of best-of-breed approach to portal usage. To answer the question it is necessary to look closer into the different use-cases and their specific advantages and disadvantages.

On an abstract level we can differentiate 5 different level of front-end integration:

1. Portal component development: This option addresses the types of development support for the integration of content coming from the “other world” into a portal.
2. Portlet syndication: Portlets (re-usable portal components) are offered by one portal (producer) and integrated into another portal (consumer) using an open standards protocol like WSRP
3. Portal component embedding: Single portal components are displayed in another portal.
4. Portal embedding: One portal is embedded in another.
5. Portal launch: A link in one portal launches the other portal in a new browser window.

Each of these types of integration is possible in one way or another. Each type also has advantages and disadvantages. Subsequently we will offer you some deeper insight and guidance.

### Portal Component Development

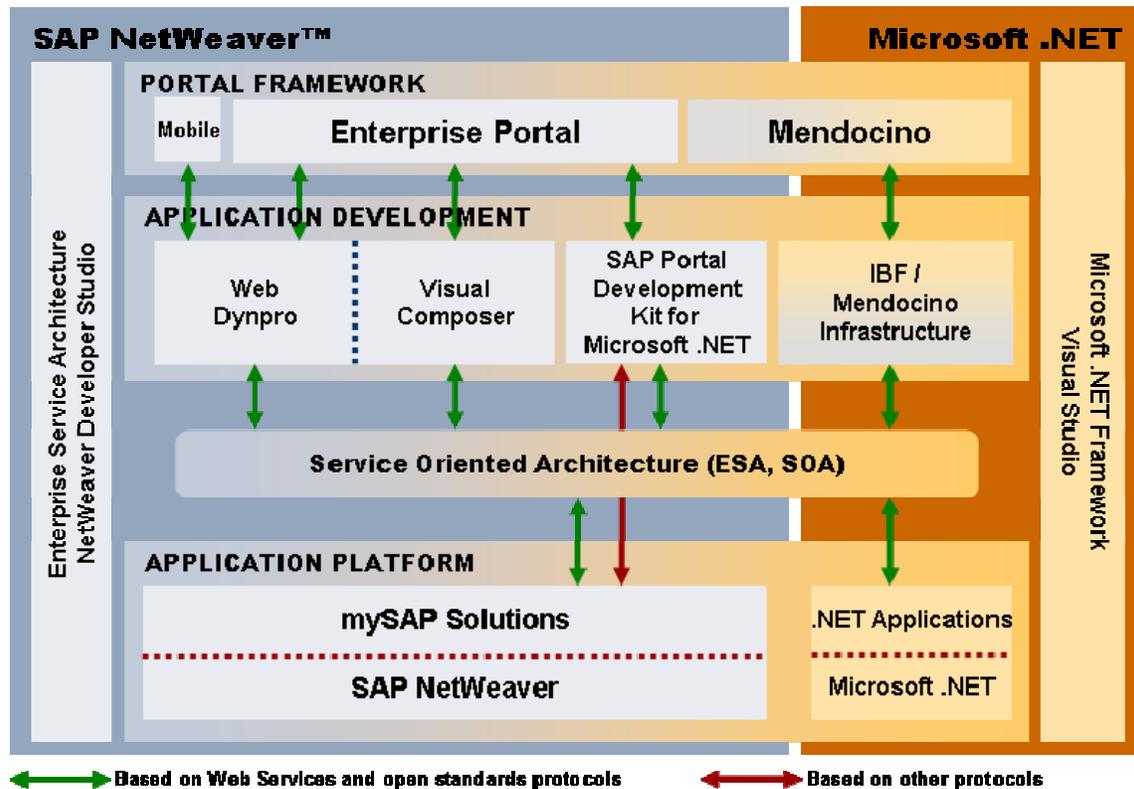
By developing custom portal components the deepest integration can be achieved. Development tools offer full access to the underlying portal platform using comprehensive class libraries and frameworks. Through web services and other open protocols virtually any functionality and process can be reached, also on other platforms. At the same time the best usability integration can be offered to the end-user because the portal component is based on the UI technology of the host portal and has the same look-and-feel as other portal components on the page.

Main disadvantage is the fact that each portal component will have to be developed and maintained individually. While perhaps 10 custom written portal components may be possible without dedicated funding, 100 or even more are not. It quickly becomes commercially unacceptable to re-implement the user interface of an entire application, while it perhaps already exists for the other portal. Therefore portal component development mainly makes sense if a new application has to be developed or if an existing application can be migrated with a large degree of code reuse.

SAP and Microsoft offer dedicated tools for portal component development on the .NET framework. Because SAP Enterprise Portal is based on J2EE SAP also offers a comprehensive development infrastructure on J2EE but this is not subject of this paper.

Because the different development paths are not always clear, we have added here two architecture pictures which will give you an overview of the existing tools and technologies available for application development from SAP and Microsoft. The first picture presents a SAP centric, the second one (further down) a Microsoft centric view. Common to

all presented development paths is that they strongly rely on a Service Oriented Architecture (SOA).



Picture 7: Development Paths for SAP-centric Applications

### Portal Development Kit for Microsoft .NET (PDK for .NET)

Portal Development Kit for Microsoft .NET is a set of tools that enables Microsoft Visual Studio .NET developers to build iViews for SAP Enterprise Portal. This way new .NET applications can be built which offer an iView user interface. Because .NET iViews programmatically are very similar to standard ASP .NET web forms it is fairly easy to bring existing ASP .NET applications to SAP Enterprise Portal.

The development process itself takes place purely in the .NET world. The developer has access to the SAP EP infrastructure through portal services and objects which are exposed as .NET objects. No Java skills are required.

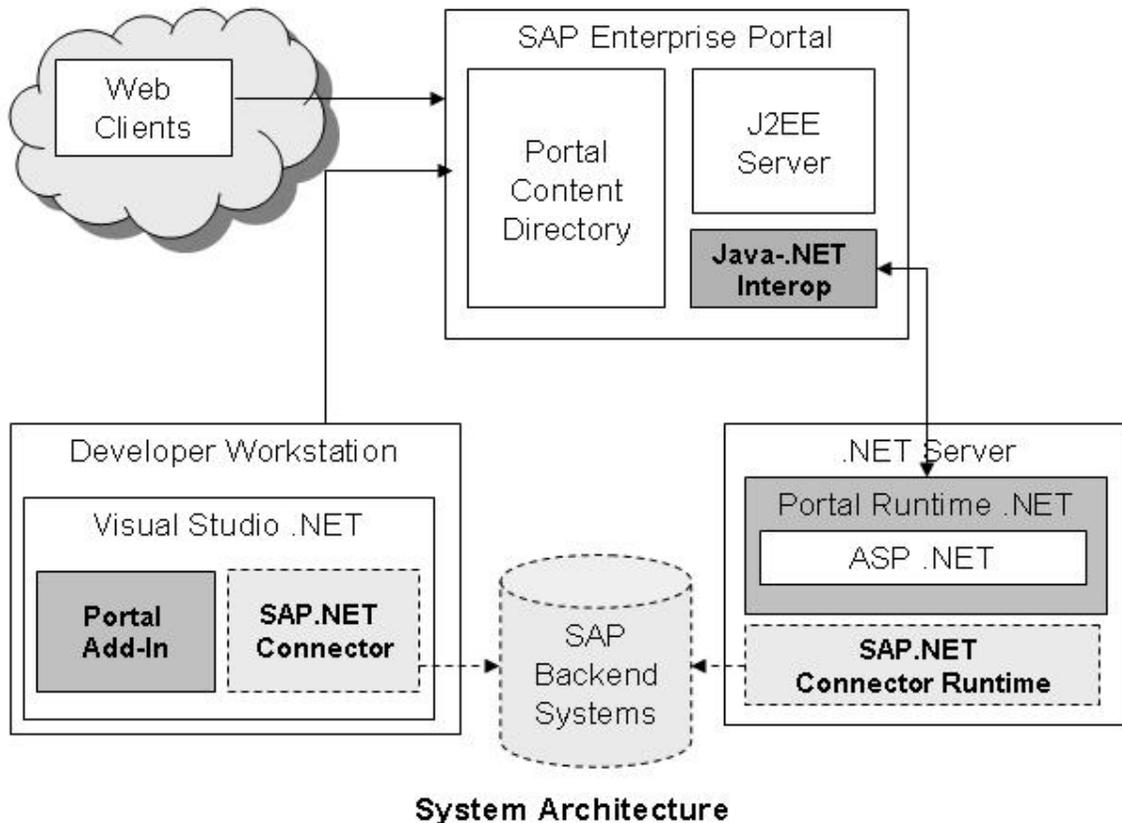
Also the PDK comes with a number of SAP .NET UI controls. These controls again are .NET controls for the developer but in the UI they appear and behave like SAP EP controls. To complete the picture the PDK .NET comes with comprehensive support for debugging, load balancing and so on.

The PDK .NET is particularly valuable because it offers a deep integration between .NET applications and SAP EP far beyond a pure visualization while protecting existing investments in Microsoft .NET technology.

Migrating the user interface of existing .NET applications can be done with limited effort because a lot of the existing coding can be re-used.

The PDK for .NET package is comprised of the following components:

- **Portal Add-in for Visual Studio 2003 (Portal Add-in)**  
An extension of Microsoft Visual Studio 2003 that allows .NET developers to build, deploy and debug portal components for SAP Enterprise Portal. For a .NET developer it takes just a few hours until he can write .NET iViews.
- **Portal Runtime for Microsoft .NET (.NET Runtime)**  
To process .NET portal components at runtime, the portal employs Portal Runtime for Microsoft .NET, an NT service that hosts ASP.NET Engine.
- **Java-.NET Interoperability Framework**  
A mechanism that enables API calls between the Java stack and .NET stack. It allows invoking Java-based portal services from .NET code, and permits SAP Enterprise Portal to pass requests to the Portal Runtime for .NET. It is implemented by the two components on the portal side, and .NET Remoting on the .NET side.



Picture 8

The PDK for .NET can be downloaded at:

<https://www.sdn.sap.com/sdn/developerareas/dotnet.sdn>

## ***Web Part Development using Visual Studio***

By using the SharePoint SDK it is possible to write custom Web Parts in order to integrate with additional application needs. By using this technology it is possible to integrate SAP content and data in a customized UI by invoking SAP Web services or using the SAP connector for Microsoft .NET.

A solution developer for SharePoint Products and Technologies will only need to know one user interface technology (ASP.NET Web pages and controls), the .NET framework and the SharePoint Products and Technologies object model to create advanced sharing solutions with SharePoint. The preferred IDE for developing customized Web Parts is Microsoft Visual Studio. Thus developers can exploit the full ASP.NET feature set. A Web Part Page is an ASP.NET file (.aspx). Web Parts are derived classes from the ASP.NET Base Class System.Web.UI.Control. The development scenarios reach from customizing Web Part pages to developing completely new Web Parts with custom properties with access to all kind of data sources within the enterprise. The requirements for developing Web Parts are:

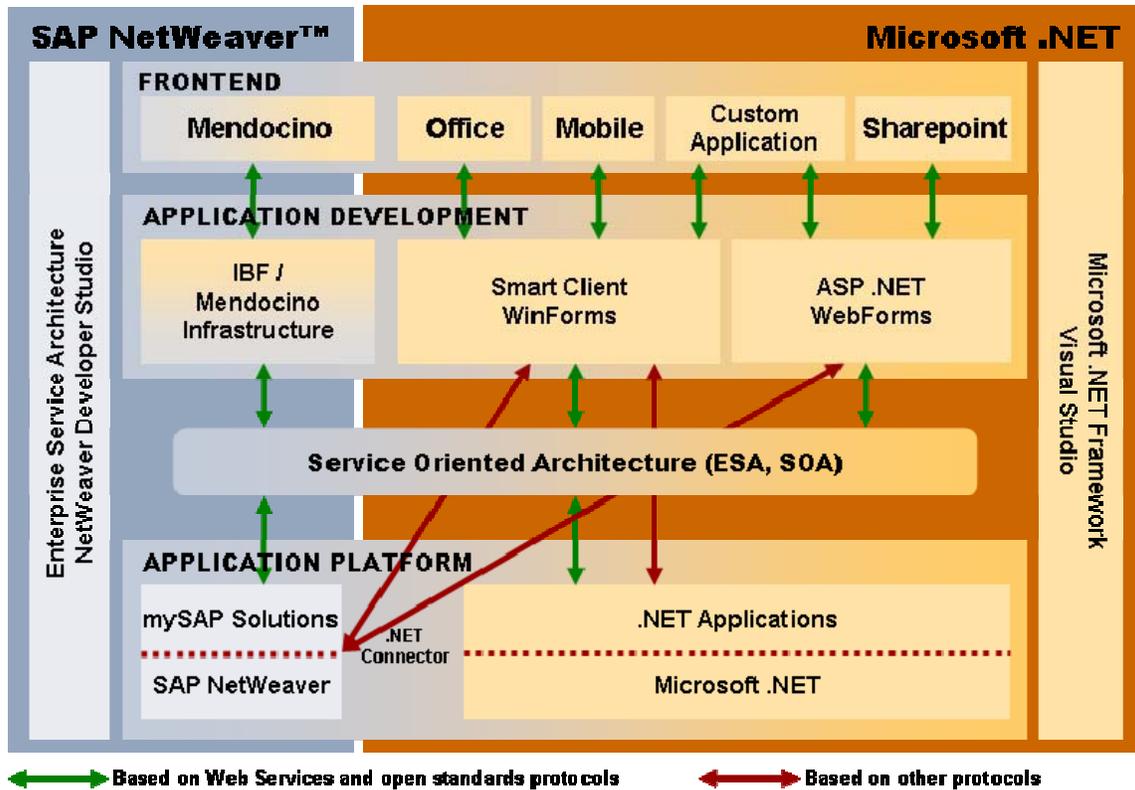
- Microsoft Visual Studio IDE
- Windows SharePoint Services
- the Web Part Library Template, which can be downloaded from [MSDN](#).

From within Visual Studio the developer has full access to Web services offered by SAP either through SAP Exchange Infrastructure (XI), via Web services on the Web Application Server or (for older SAP releases) through the SAP connector for Microsoft .NET.

Migrating the UIs of existing SAP applications is rather difficult because they are developed in either ABAP or Java. The re-use of existing code therefore is not possible.

The SharePoint SDK is available at:

- <http://www.microsoft.com/downloads/thankyou.aspx?familyId=AA3E7FE5-DAEE-4D10-980F-789B827967B0&displayLang=en>
- [http://msdn.microsoft.com/library/default.asp?url=/library/en-us/spptsdk/html/CreateABasicWP\\_SV01003709.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/spptsdk/html/CreateABasicWP_SV01003709.asp)



Picture 9: Development Paths for Microsoft-centric Applications

### Web Services based integration

In a Web services architecture it will become possible that virtually all applications in a system landscape of an organization can communicate with each other using open standards protocol. In a joint announcement in 2004 Microsoft and SAP expressed their "...shared commitment to Web services as the foundation for the next generation of enterprise software".

Technically defined, Web services are self-contained, modularized, executable entities that can be published, searched for, and accessed across a network. According to this definition there are in general three objects involved into the Web services based communication:

- **Publisher**  
An organization can expose some processes as Web services and publish the service description to a directory
- **Service directory**  
A Web service directory contains descriptions of published services and can be browsed either by a human reader or by an automatic agent
- **Consumer**  
A person or organization that wishes to access the published Web service

There is a plenty of resources available on the Internet related to Web services. You can find valuable information on

- <http://www.w3.org>
- <http://www.oasis-open.org>
- <http://www.sdn.sap.com>
- <http://msdn.microsoft.com>

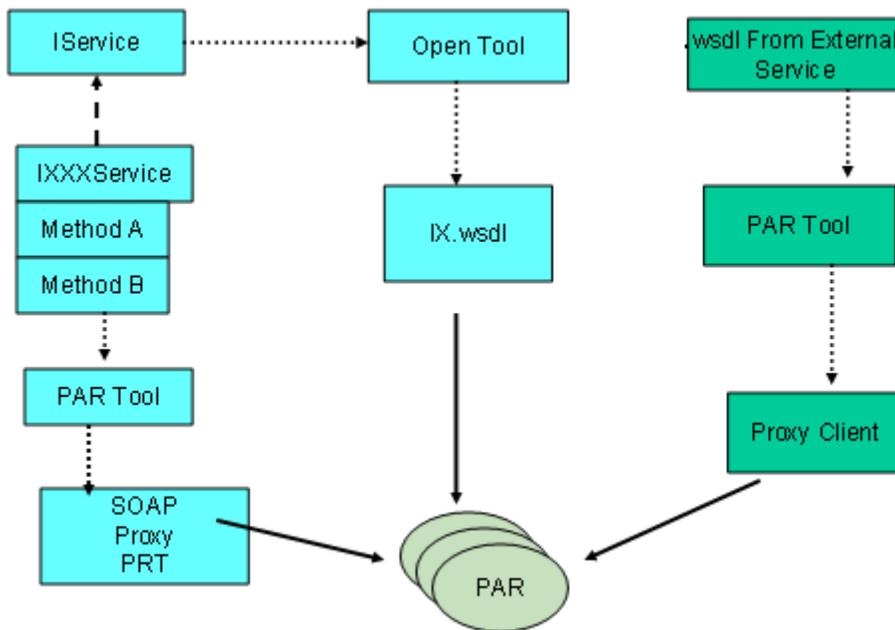
Both SAP and Microsoft products offer comprehensive support for publishing and consuming Web services. On this basis we want to point out some technical possibilities of implementing Web service based integration between SAP Enterprise Portal and Microsoft SharePoint Technology.

### **Web Service Support by SAP Enterprise Portal**

Web services technology is a building block of the concept of portal services. Portal services can be transformed automatically into standard Web services to expose their interfaces to external and internal components using the SOAP protocol.

In the same way, the Portal Runtime (PRT) provides tools and facilities to generate proxies to external Web services. This means that portal components and portal services can easily access external Web services to build a portal application.

The following figure shows the two aspects of the Web services support in the PRT. The central piece of the architecture is the NetWeaver Developer Studio plug-in which is used either to generate a WSDL from an existing service or a Web services client from a WSDL. The result of both generations is packaged in a PAR file.



Picture 10

The Portal Runtime Technology supports following Web services related standards:

- SOAP 1.1 specification without an attachment
- WSDL 1.1 specification

More information about Web services support can be found in the *SAP Enterprise Portal Developer Guide* as part of the SAP NetWeaver Developer Studio Documentation.

### **Web Service Support by Windows SharePoint Services**

Being .NET applications, Windows SharePoint Services provide a rich support for Web services embracing the ability to consume remote or local Web services and the ability to expose SharePoint functions through Web services.

Windows SharePoint Services deliver several Web services as one of two options to use its object model. Compared with another one – class library DLLs – Web services do not provide as much functionality but they can be used to implement custom applications that run anywhere on the network.

Following Web services are delivered by default with Windows SharePoint Services:

<b>Service</b>	<b>Description</b>
Administration	Provides methods for managing a deployment of Windows SharePoint Services, such as for creating or deleting sites.
Alerts	Provides methods for working with alerts for list items in a SharePoint site.
Document Workspace	Exposes the Document Workspace Web service and its eleven methods for managing Document Workspace sites and the data they contain.
Forms	Provides methods for returning forms used in the user interface when working with the contents of a list.
Imaging	Provides methods that enable you to create and manage picture libraries.
List Data Retrieval	Provides a method for performing queries against lists in Microsoft Windows SharePoint Services.
Lists	Provides methods for working with lists and list data.
Meetings	Provides methods that enable you to create and manage Meeting Workspace sites.
Permissions	Provides methods for working with the permissions for a site or list.
Site Data	Provides methods that return metadata or list data from sites or lists in Microsoft Windows SharePoint Services.
Sites	Provides a method for returning information about the site templates for a site collection.
Users and Groups	Provides methods for working with users, site groups, and cross-site groups.
Versions	Provides methods for working with file versions.
Views	Provides methods for working with views of lists.
Web Part Pages	Provides the methods to send information to and retrieve information from XML Web services.
Webs	Provides methods for working with sites and subsites.

(Source: Microsoft SharePoint Products and Technologies 2003 SDK)

While documents in document libraries already will become available to SAP Enterprise Portal via the announced WebDAV Connector for SharePoint, a possible case of Web service usage could be the access to SharePoint lists from SAP Enterprise Portal. It would be quite easy to develop iViews which retrieve list data using Web services.

For more details on programming Web services for Microsoft Windows SharePoint Services please review [Microsoft SharePoint Products and Technologies 2003 SDK](#).

## Portal Component Syndication

It would be great if one could just drag-and-drop a Web Part into Enterprise Portal or to just use an iView inside of SharePoint, wouldn't it? Well, it's not quite that easy. Simply because WSS and EP are quite different products based on different concepts. Still also in this area there are standardization efforts underway which may make a lot of this drag-and-drop approach possible. The two main standards today are WSRP and JSR168. It is not likely that Microsoft will support JSR168 as it is based on a Java Community Process. WSRP on the other hand explicitly is designed to be implemented in .NET environments as well. Here is the status quo.

### *Web Services for Remote Portlets (WSRP)*

Web Services for Remote Portlets is a web services protocol for aggregating content and interactive web applications from remote sources. This means that WSRP permits the re-use of portal components from one portal (producer) in another portal (consumer) based on web services. According the definition in Wikipedia "Portlets are reusable Web components that display relevant information to portal users". Ideally SharePoint Web parts could be plugged into Enterprise Portal and iViews into SharePoint. Both Microsoft and SAP participate actively in this Technical Committee of the OASIS consortium (see also [http://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=wsrp](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrp)).

While well-defined standards make interoperability smoother in many cases, the WSRP protocol currently offers only limited benefits to users of EP and WSS/SPS. There are two reasons for this limitation:

- The WSRP protocol is currently released with version 1.0, which has some limitations regarding (besides others) security management and content hierarchies. The main limitation however, is that WSRP is aiming towards the portlet level and not the page level and therefore does not offer server-side eventing which would be crucial for that purpose.. WSRP 2.0, which is not yet finalized, will offer significant improvement but still would have to be implemented.
- EP and WSS both offer specific features as part of their infrastructure. Examples for EP are drag-and-relate, unification. Examples for WSS/SPS: joining of lists. Some of these may never be available through WSRP.

Partially due to these limitations today there is only very limited support for WSRP 1.0:

- SAP Enterprise Portal will add support for WSRP 1.0 as part of NetWeaver 04s as both producer and consumer. The planning for WSRP 2.0 depends on the finalization of this new version of the protocol.
- Microsoft supports WSRP (consumer and producer) through Web Services facades in Visual Studio. Today Microsoft supports WSRP 1.0 through an unsupported shared-source Web Part for consuming services inside of SPS. For producing WSRP Microsoft offers a programming example. This means the preferred approach for customers, who want to leverage WSRP for SharePoint today, is through custom development using web services in Vis-

ual Studio.

Further information about Microsoft WSRP:

- <http://www.gotdotnet.com/Workspaces/Workspace.aspx?id=805b3559-c810-4119-86f4-11ba5c16a5b0>

## Portal Component Embedding

A fairly primitive approach to cross-integrating portal content is to display Web Parts or iViews into the opposite product. This is done by creating an empty iView or Web Part and referring to the URL of a Web Part or iView from the other portal. Upon execution this one will be displayed to the user as part of a portal page. This type of integration can also be called “URL integration” or “visual integration”.

In some cases this may be a good enough solution, mainly when the targeted portal component works quite isolated and does not contain links. Most portal components in real-life applications however, will not look good or simply won't work at all. The reasons for these problems are quite clear:

- The portal component is taken out of its survival system and ‘transplanted’ into an unknown environment.
- The visual appearance is the one of the source portal and by default will not blend into the destination. This can be partially fixed by modifying the layout (for instance cascading style sheets).
- The usability is different because UI controls look and behave differently. This can't be fixed.
- Services from the home portal, like navigation, user management and eventing between portal components, are inaccessible.

Please also note that embedding of portal components generally makes it necessary to purchase the appropriate licenses for both the hosted and the hosting portal.

As you can see portal component embedding is not really a recommended way of bringing content to other portals. Nevertheless we want to show you how you can do it.

Generally the following criteria should be taken into consideration when deciding which integration scenario makes sense.

- Clearly define which UI elements should be integrated
  - integration of an entire portal site by URL
  - integration of single portal components or portal component pages by URL linking
- Is a customization concerning colors and other surface elements desired?
- Is a special behavior necessary concerning embedded links within the portal elements?

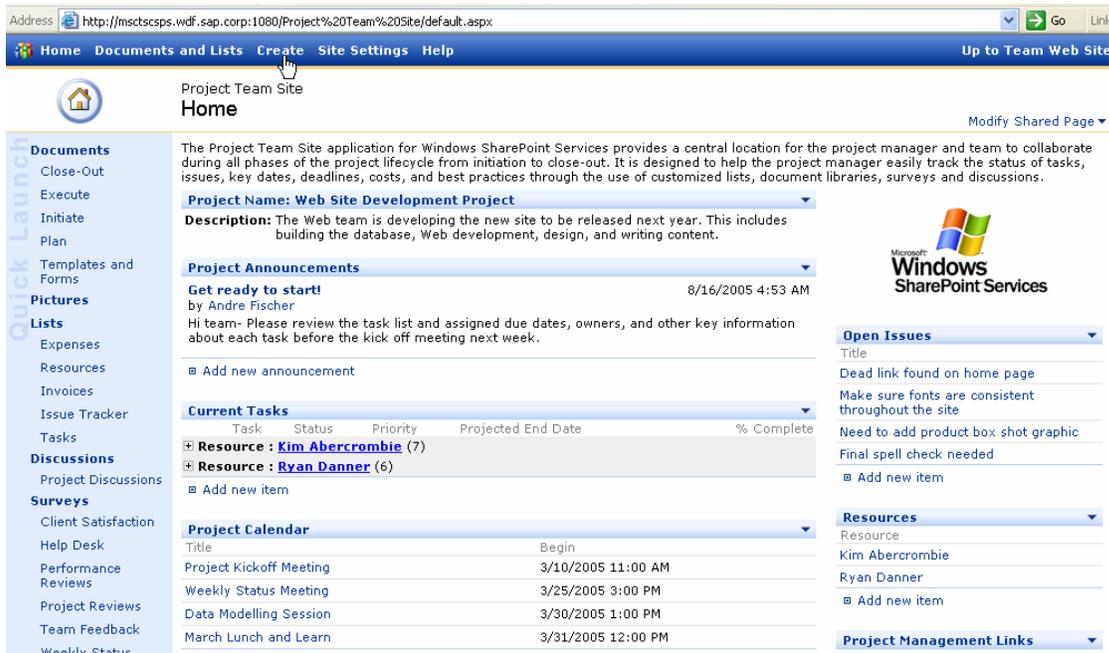
### ***Embedding customized Web Part pages or lists into SAP Enterprise Portal***

In order to understand following chapters, you need to have a rough understanding of the concepts and terminology used in Windows SharePoint Services. For this reason, we recommend the technical article [Architectural Overview of Windows SharePoint Services](#) on MSDN.

Here, we try to give a short explanation for terms used below:

- *Web Parts* are the basic building block of the user interface in a SharePoint solution. Windows SharePoint Services comes with a number of Web Parts, and many more are available in the online [Web Part gallery](#). Compared to the SAP Enterprise Portal, Web Parts are counterpart to iViews.
- *Web Part Pages* contain one or more Web Parts. The layout of Web Parts on the corresponding page is defined by Web Part zones. This concept is similar to SAP's portal pages that contain one or more iViews in a defined layout.
- *Lists* are table-like UI elements that allow SharePoint users to easily collect and share information. There are a number of built-in lists delivered with WSS. Additionally, libraries are a special form of lists used to store documents, pictures or forms. In addition the user is able to create lists of his own.

As a rule, a Web Part page contains several Web Parts as you can see on the screenshot below.



Picture 11

The procedure required to implement this scenario comprises adjustments of SharePoint pages and creation of default URL iViews and other corresponding objects like sites, worksets or roles in SAP EP.

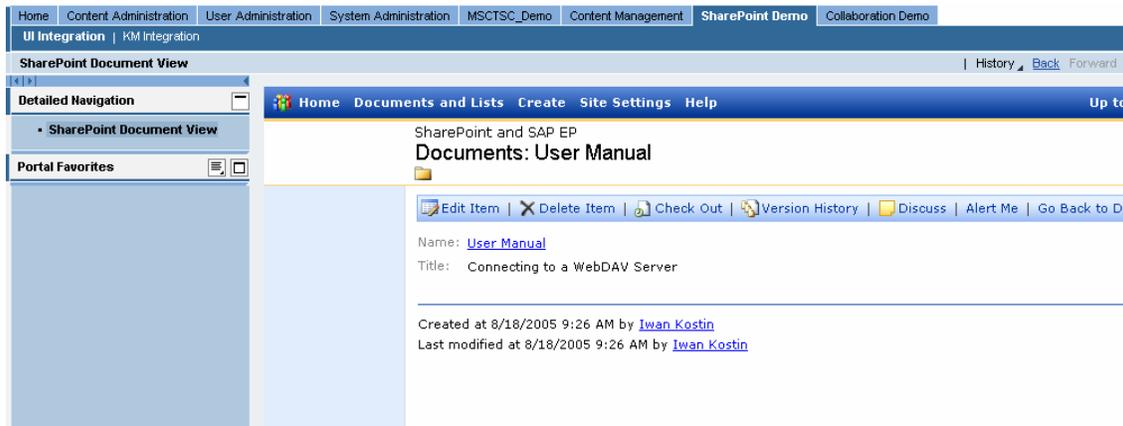
Web-based URL iViews in the SAP Enterprise Portal provide a simple way to create iViews that retrieve content directly from a Web page or Web-based application. Technically, a URL iView is a collection of meta attributes. The URL targeting the information source is one of them. The wizard for creation of an URL iView provides a built-in browser enabling you to navigate within a Web site in order to retrieve the URL of the source Web page.

**Note:**

Currently, the built-in browser does not support Web sites that require authentication. As a workaround you can use an external browser window to determine the URL of the targeted Web page.

For the detailed description on how to customize Web part pages and to create URL iViews in SAP EP please read the chapter *Integration of a SharePoint List View into SAP Enterprise Portal* in the How to section.

The described approach to integrating individual Web Parts has the advantage of a better usability compared with the integration of complete Web Part pages that contain several Web Parts as described below.



Picture 12

In this picture we can see the navigation area and banner of both the host and guest portal. You can only avoid this behavior if you integrate Web Parts without links or if an embedded link opens the target page in a new browser window.

Nevertheless please keep in mind the weak point of this approach. Clicking links embedded in SharePoint views may cause the opening of further SharePoint pages inside of the iView area including its own banner and navigation. Such an iView in most cases will not be usable any more.

***Capturing SharePoint content into SAP Enterprise Portal***

Similar to the procedure discussed in the previous chapter, this chapter presents an approach to display particular SharePoint content within an Enterprise Portal page. The main difference between both approaches is the customization effort on the SharePoint side.

This procedure takes advantage of the built-in capture tool provided by SAP Enterprise Portal. The capture tool allows you to choose to display either the entire Web page or a selected area within it. Therefore, no customization has to be made by SharePoint administrators.

You can find a step-by-step explanation for creating an appropriate URL iView in the chapter *Web Parts integrated in SAP Enterprise Portal* in the How-To section.

**Note:**

To determine the page area you want to display at the runtime, you use the built-in browser. Unfortunately, the built-in browser does not support Web sites that require you to logon. In contrast to the integration approach demonstrated in the previous section, there is no workaround.

All things considered, you can implement this scenario only for anonymous SharePoint sites, which contradicts the philosophy of SharePoint.

Weak points discussed in the chapter *Integration of customized Web Part pages into SAP Enterprise Portal* apply to this scenario, too. Although you could avoid some problems with iView parameters *Open iView links in new window* and *Remove client events from Web site content*.

### ***SAP iView Web Part Toolkit for SharePoint Products and Technologies***

Microsoft offers the unsupported SAP iView Web Part Toolkit for SharePoint Products and Technologies, which enables SharePoint administrators to display SAP Portal iViews within Microsoft Windows SharePoint Services/SharePoint Portal Server. It was designed for scenarios with SAP Enterprise Portal and SharePoint running in parallel, and the need to expose selected iViews also in SharePoint.

The toolkit contains a custom Web Part, called the *iView Web Part* that provides a window onto a running iView on a SAP Portal page. In addition it contains two executables which extract metadata from an SAP portal page and generate a Web Part page.

The toolkit automates the process of embedding the iView into WSS. It does not fix the general issues of portal component embedding.

Signing the user on to the SAP Portal is handled by simulating the sign-on dialog on behalf of the user with an additional HTML page. This page will be used whenever an SAP SSO2 token is not available to the browser. This mechanism does not resemble a true single sign-on but a simple user-mapping with all its drawbacks. For instance password changes have to be repeated in all involved systems.

The SAP iView Web Part Toolkit is made available under the Shared Source License at the following URL:

<http://www.gotdotnet.com/workspaces/workspace.aspx?id=d6129dc0-efc4-457d-a821-fd26aef566dec>

### **Portal Embedding**

With this approach an entire guest portal is displayed in a host portal based on its URL. Though technically very simple to implement, from a usability perspective this type of integration is the worst case scenario:

- The guest portal brings its own banner and navigation. This takes up a lot of space on the screen and on smaller screens may severely impact the usability for the end-user.
- Host portal and guest portal have a different look-and-feel. Because visually they are presented next to each other the user quickly gets confused.

- Each time the guest portal is displayed, the server has to instantiate a session and to authenticate the user. This process is very resource intensive.

In case of such a loose integration between SAP Enterprise Portal and Microsoft SharePoint, it would be more recommendable to open the SharePoint site in a new browser window (see chapter Portal Launch).

## SharePoint sites embedded in SAP Enterprise Portal

Embedding complete SharePoint sites without any changes does not require any effort in SharePoint and only includes default steps needed for creation of URL iViews in SAP EP.

We provide a detailed description of this procedure in the chapter *SharePoint sites integrated in SAP Enterprise Portal* and in the How to section below.

### Note:

Usually, SharePoint sites require user logon. In its current version the built-in browser does not support such Web sites. Therefore, you need to use an additional browser window to determine the target URL and required URL parameters, and enter them manually using the URL iView Editor.

## Portal Launch

This type of integration is not really integration anymore. Referencing external Web based applications or Internet sites is a basic functionality of the Web and certainly can be used in an integration scenario between SAP Enterprise Portal and SharePoint.

A SAP EP content administrator can provide links to external applications in several ways:

- Embedding a URL iView as a navigation node in one of the portal's navigations. The way of doing it was already used to embed SharePoint sites into the SAP EP. The only difference is the iView property *Launch In* that you have to set to a value of "1" to display the started iView in an external window.
- You can reference SharePoint sites or any other Web based applications by creating external links in KM repositories. Actually, providing links this way has the nice side-effect that these links are indexable. That means user would see them in search result sets when they match the search expression.

One should keep in mind when opting for Portal Launch integration that the user is forced to change the environment with all related consequences like different look-and-feel and the need to switch between several windows. A benefit for the user in this scenario is that standard single sign-on mechanisms should make additional logins obsolete.

## Collaboration Interoperability

Both Enterprise Portal and Sharepoint offer extensive collaboration functionality which, for obvious reasons, integrates nicely with all other parts of the respective portal's infrastructure.

Here is a brief description of core features available to both products:

Benefit	Description
Collaboration rooms, Team Sites	Create, maintain, and manage virtual work spaces for teams, workgroups, and communities. Invite new team members and whole groups, provide access rights, share knowledge, ideas, and work in a team-oriented environment. Share all project-related tools, services, and information sources from a single point of access.
News, discussions, and folders	Edit and publish team and project news to all members. Share information and exchange ideas – all organized by discussion topics. Because collaboration capabilities are tightly integrated with the document repositories, team- and project-related documents can be managed with functions for check in, check out, locking, versioning, notifications, and folder management.
Team calendar	Schedule meetings and synchronize calendars by integrating with groupware solutions like Microsoft Exchange Server.
Task assignment	Create, browse, assign, edit, and track team tasks.
Instant messaging and chat	Create and send instant messages to one or more persons for immediate response. Hold real-time discussions with multiple users.
Contact list	See whether other portal users are online and available. Start collaboration services directly from the contact.
Application sharing	Share multiple applications, documents, and your desktop and portal components in real time from any portal page. Make application sharing an integral part of meetings that are facilitated by external real-time communication tools.

Similar to what can be said about the main portal products, each of the collaboration tools comes with a different focus. While SAP NetWeaver Collaboration fosters a good integration with all kind of applications inside of SAP Enterprise Portal, SharePoint also here benefits from a deep integration with Microsoft Office.

The question remains, do SAP NetWeaver Collaboration and SharePoint collaboration also integrate with each other? The brief answer is “No, there is no dedicated functionality available that immediately leverages the other solution’s collaborative data”. The longer answer could be “No, and there is no real need for direct interoperability on that level”.

The end-user will use the collaboration functionality available in the portal that he spends most time in. A lot of the data makes only sense in that immediate context and is not really needed anywhere else (team news, discussions, contacts...). Common to both solutions is that they synchronize calendar information with Exchange Server.

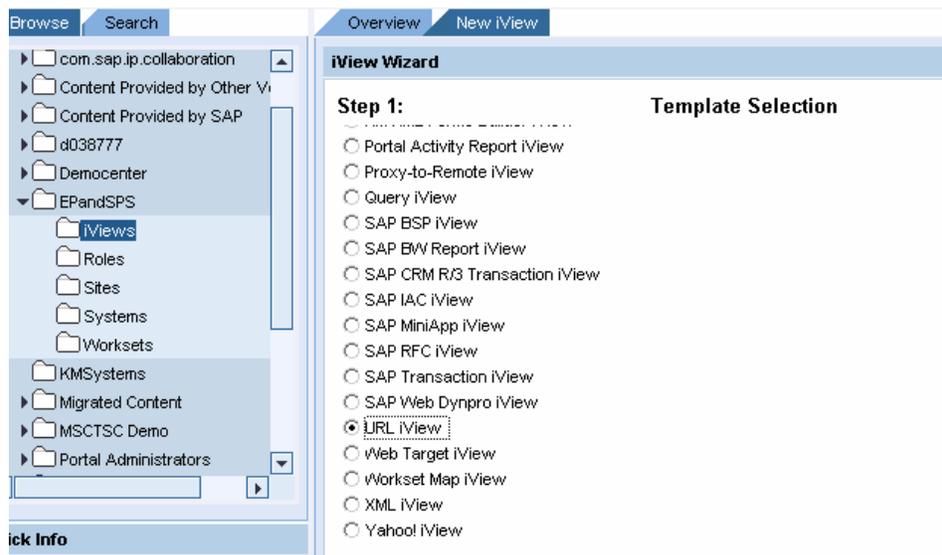
## How-To

### How-To: Simple URL Integration

#### *SharePoint sites integrated in SAP Enterprise Portal*

Below is the procedure for creating URL iView. This procedure is not SharePoint specific and can be used for other Web sites or Web-based applications:

1. Launch the iView Wizard by right-clicking the appropriate folder and choosing from the context menu that appears *New -> iView*.
2. On the *Template Selection* screen, you initiate the URL iView wizard by choosing the *URL iView* Template:



3. You define the basic properties of the iView on the *General Properties* screen:

The screenshot shows the 'iView Wizard' interface with 'Overview' and 'New iView' tabs. The main heading is 'Step 2: General Properties'. It contains several input fields: 'iView Name \*' with the value 'SharePoint and SAP EP Site'; 'iView ID \*' with the value 'spsapepsite'; 'iView ID Prefix (Example: com.companyname)' which is empty; 'Master Language \*' with a dropdown menu set to 'English'; and a 'Description' text area which is empty.

4. In case, you have a SharePoint site which requires user authentication determine the target URL in an external browser window:

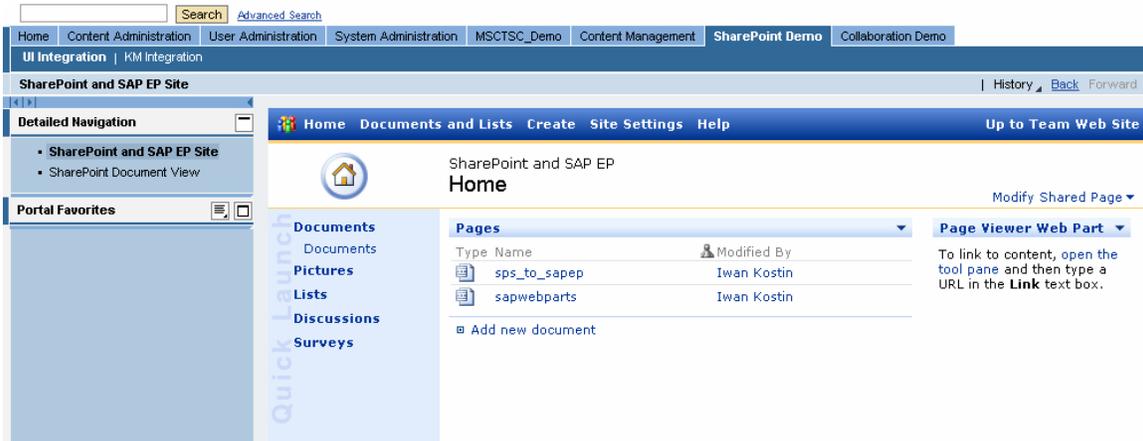


5. Enter the Web page URL in the *Enter URL* box:

The screenshot shows the 'iView Wizard' interface with 'Overview' and 'New iView' tabs. The main heading is 'Step 3: Define Source URL'. Below the heading is explanatory text: 'Allows you to create an iView that accesses a URL address on the Web. The iView displays the content provided at' and 'You can display an entire Web page or capture a selected portion.' There is an 'Enter URL \*' label above a text input field containing the URL: <http://msctscsps.msctsc.sap.corp:1080/spsandsapep/default.aspx>. To the right of the input field is a 'Browse/Capture' button. A note at the bottom states: 'Note: after the wizard has finished, you can define and edit parameters associated with the URL, and apply the look & feel in the iView Editor.'

- Confirm your entries without using the Browse/Capture capabilities of the built-in browser if you specified a SharePoint site which requires user logon.

The result looks like on the following below:



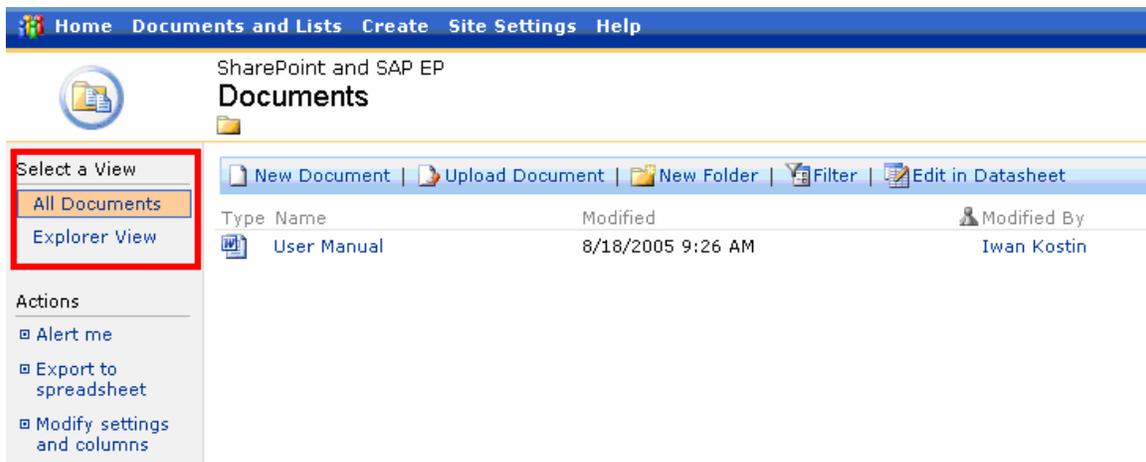
### ***Integration of a SharePoint List View into the SAP Enterprise Portal***

One of the most common tasks when integrating SharePoint UI elements into SAP Enterprise Portal is using SharePoint List views within an URL iView.

The following steps describe necessary tasks to achieve this goal. First of all, the following instructions work with any SharePoint site no matter whether the user wants to use any of the lists on your site or your own customized Web Part Pages.

We want to discuss the given task with regard to a given document list. Concerning the implementation of other SharePoint Pages you will find the explaining hints mentioned in the text.

In order to integrate an existing SharePoint list view we start by calling its URL in Internet Explorer.

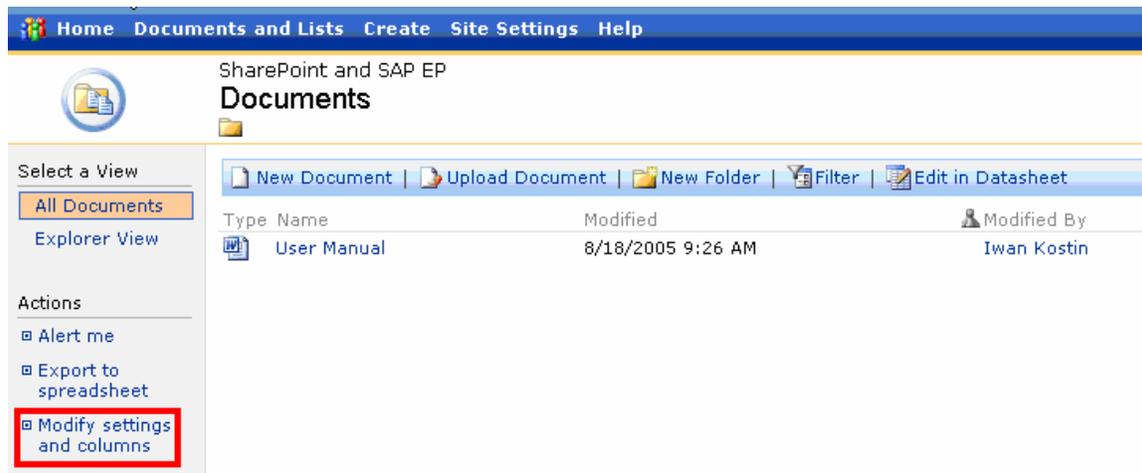


As you can see in the screenshot above, there are usually several views for one list. Views make it possible to see information contained in the list in a variety of ways.

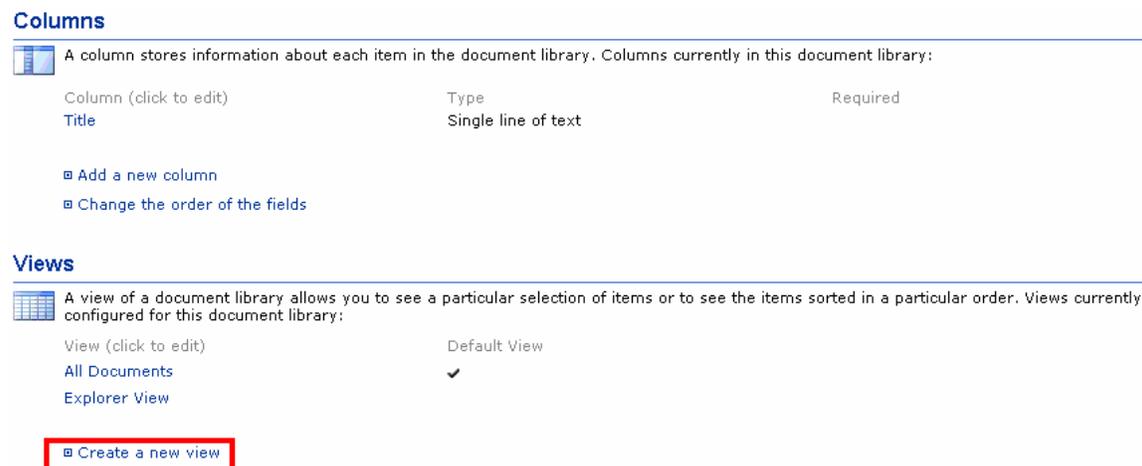
Our intention is not to change one of the existing views in order to leave the familiar look and feel for a SharePoint user untouched. Instead we will create a dedicated list view to be able to integrate it later into SAP Enterprise Portal.

The explanation below contains all required steps in SharePoint to create a new view and to modify it by means of Microsoft FrontPage followed by short description on how to integrate the customized list view into SAP EP.

To start with creation of a new view click *Modify settings and columns* on the page that display the list:



At the end of the customization dialogue we choose the *Create a new view* menu.



The *Standard View* template will suit best for our purposes right now.

## SharePoint and SAP EP Documents: Create View

Use this page to select the type of view you want to create for your data.

 **Standard View**  
View data on a Web page. You can choose from a list of display styles.

 **Datasheet View**  
View data in an editable spreadsheet format that is convenient for bulk editing and quick customization.

 **Calendar View**  
Display your data in a calendar format.

For testing purposes we only name the view, but leave all the defaults as they are. If desired the view can be customized in several ways, concerning grouping, filtering and so on. It is important, to choose the *Create a Public View* option in the audience section.

## SharePoint and SAP EP Documents: Create View

Use this page to create a view of this document library.

### Name

Type a name for this view of the document library. Make the name descriptive, such as "Sorted by Author", so that site visitors will know what to expect when they click this link. [Show me more information.](#)

View Name:

Make this the default view  
(Applies to public views only)

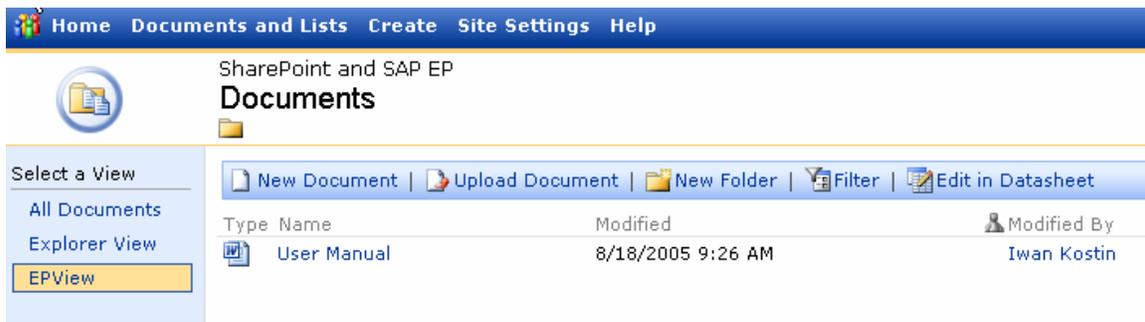
### Audience

Select the option that represents the intended audience for this view. [Show me more information.](#)

View Audience:

- Create a Personal View  
Personal Views are intended for your use only. However, if given the correct URL, others may use, modify or delete your personal view.
- Create a Public View  
Public views can be visited by anyone using the site.

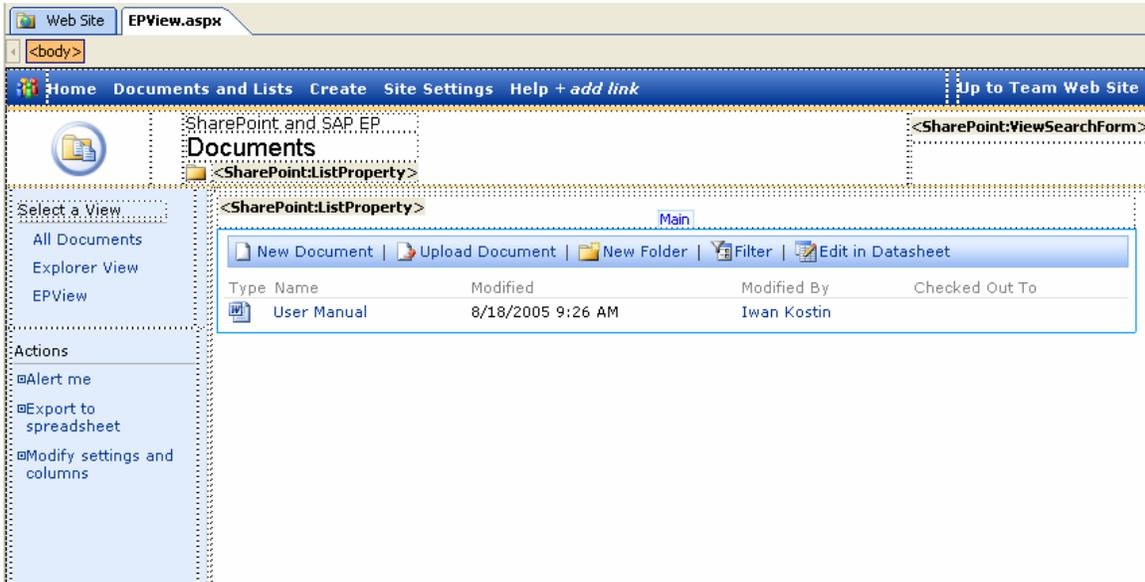
After implementing the view we invoke it.



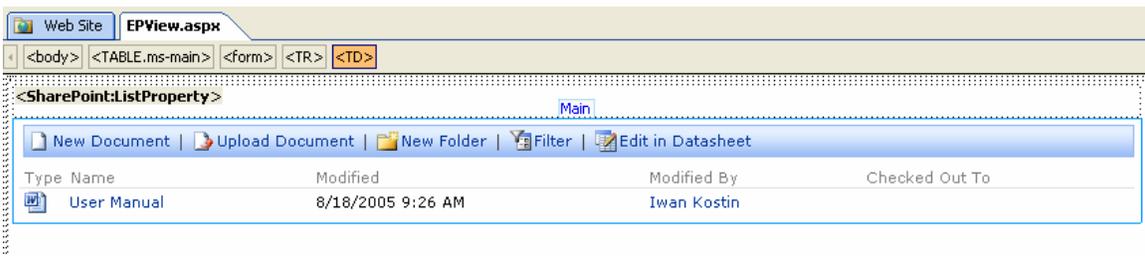
The screenshot shows the SharePoint interface for a document library named "Documents". The navigation bar includes "Home", "Documents and Lists", "Create", "Site Settings", and "Help". The page title is "SharePoint and SAP EP Documents". A "Select a View" dropdown menu is open, showing "All Documents", "Explorer View", and "EPView" (which is highlighted). The main content area shows a toolbar with "New Document", "Upload Document", "New Folder", "Filter", and "Edit in Datasheet". Below the toolbar is a table with columns "Type", "Name", "Modified", and "Modified By". The table contains one entry: "User Manual" modified on "8/18/2005 9:26 AM" by "Iwan Kostin".

For the next steps it is important to mention that Microsoft FrontPage 2003 has to be installed on the client machine. As mentioned at the beginning of this chapter any SharePoint site can be customized the way. The next remarks describe the necessary tasks.

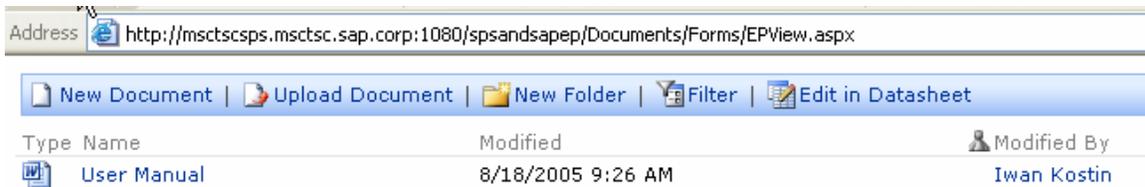
If FrontPage is installed, we will find a command in the Internet Explorer toolbar, and can open the aspx page for our view in FrontPage for editing. If your rights in SharePoint are sufficient (you have to be a member of the Web Designers group) FrontPage opens the view in edit mode.



Now we can customize the view concerning all UI elements which are not desired. We just delete them.



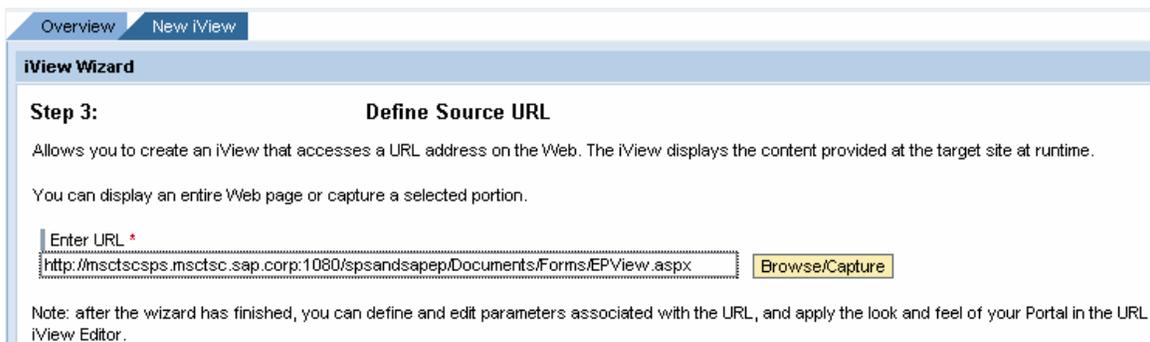
After saving the aspx page and closing FrontPage a reload in Internet Explorer shows our newly modified list view.



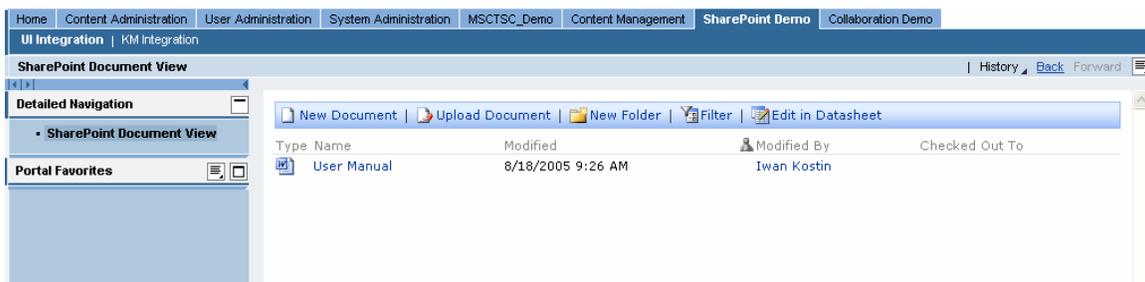
For the procedure of creating URL iView for SAP Enterprise Portal see the section *SharePoint sites integrated in SAP Enterprise Portal*. As already mentioned there, you have to determine the URL for the new document library view in an external browser window before you put it into the *Enter URL* field:



Afterwards enter the URL in the URL iView definition in Step3.



As you can see on the following screenshot, the result of the customized SharePoint view is a nearly seamless integration of the SharePoint content in SAP EP



### ***Capturing SharePoint Content into Web Parts integrated in SAP Enterprise Portal***

Since the handling of the iView Wizard is already described in the chapter *SharePoint sites integrated in SAP Enterprise Portal*, we only discuss those steps that differ from the procedure above.

We start this How to chapter with the definition of the source URL. The targeted URL might be a SharePoint Web Part page or a document library view.

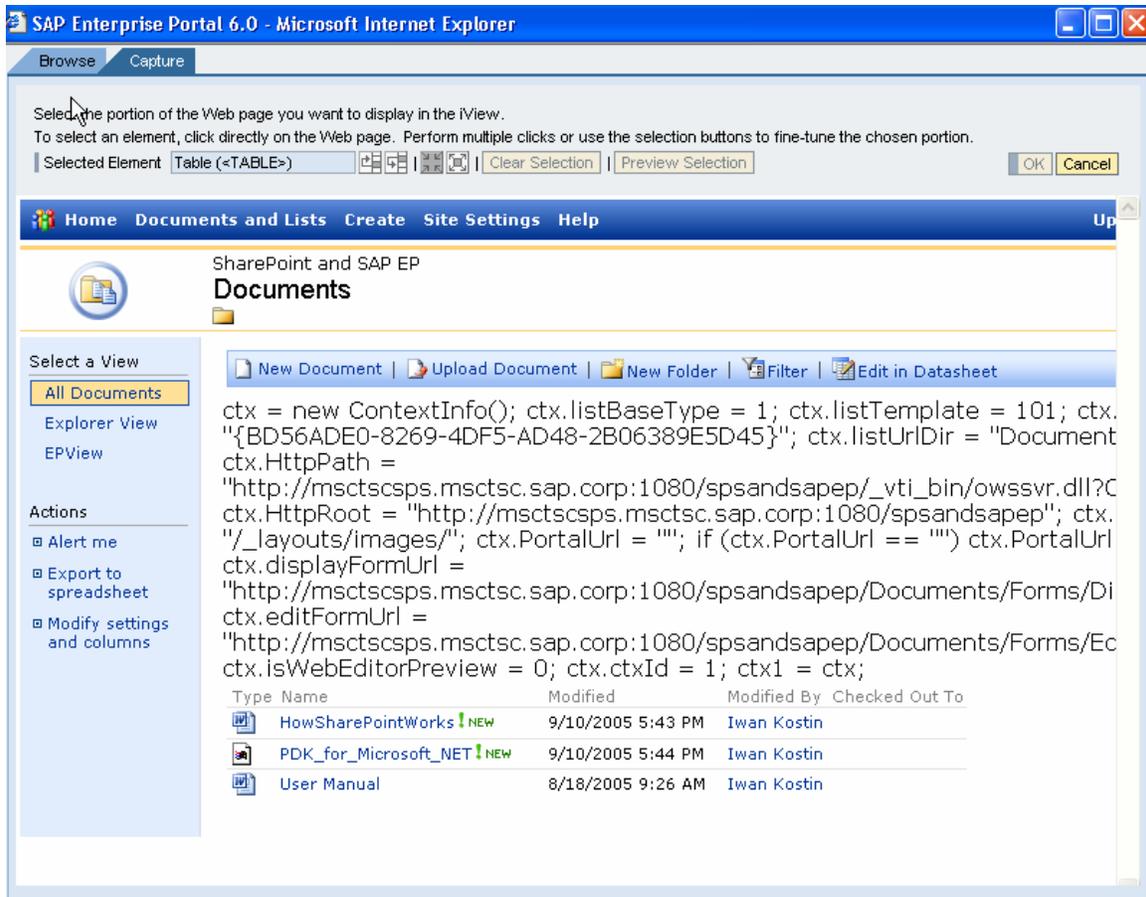
The screenshot shows the 'iView Wizard' interface with the 'New iView' tab selected. The current step is 'Step 3: Define Source URL'. The text explains that this step allows creating an iView that accesses a URL on the Web. Below the text is a text input field containing the URL 'http://msctscsps.msctsc.sap.corp:1080/spsandsapep/Documents/Forms/AllItems.aspx' and a 'Browse/Capture' button. A note at the bottom states: 'Note: after the wizard has finished, you can define and edit parameters associated with the URL, and apply the look and iView Editor.'

In the *Browse* tab, you can browse the Web to determine needed site:

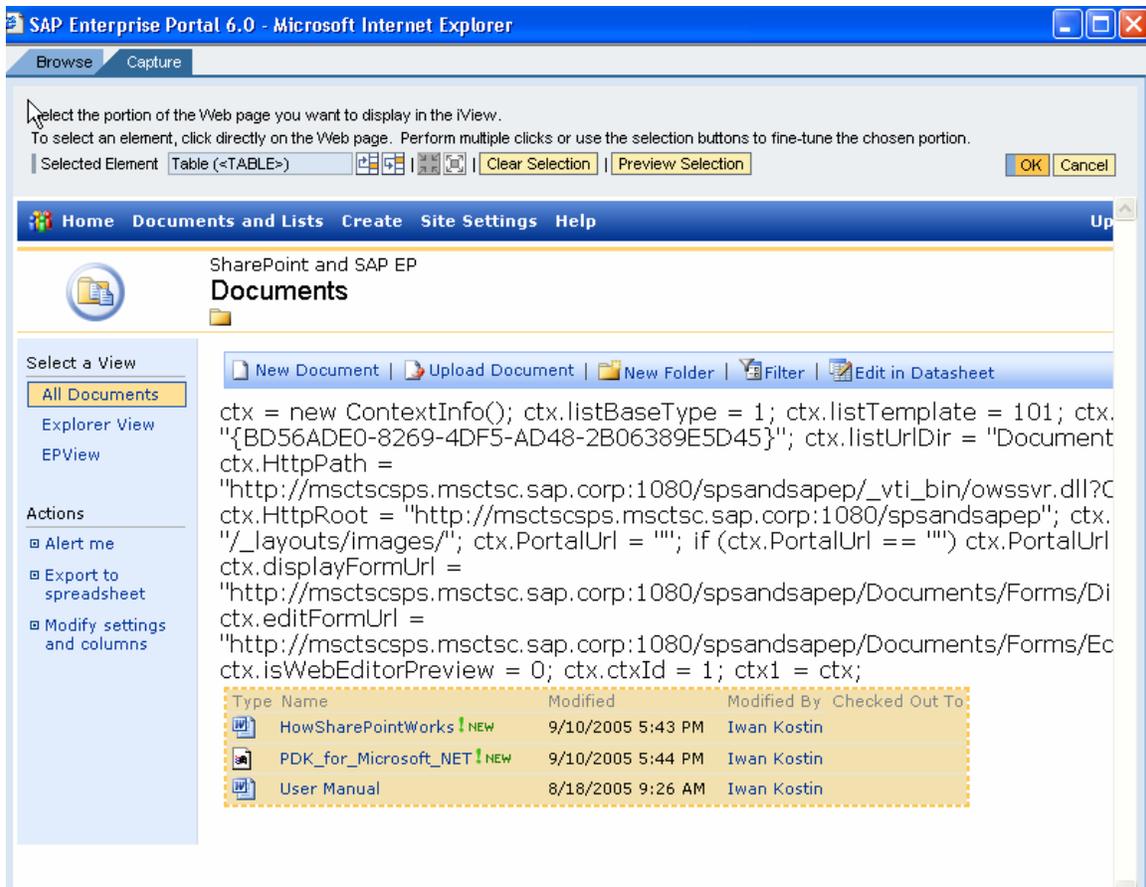
The screenshot shows a Microsoft Internet Explorer window titled 'SAP Enterprise Portal 6.0 - Microsoft Internet Explorer'. The 'Browse' tab is active. The page content includes a navigation bar with 'Home', 'Documents and Lists', 'Create', 'Site Settings', and 'Help'. Below this is a 'Documents' library view for 'SharePoint and SAP EP'. The view shows a table of documents with columns for 'Type', 'Name', 'Modified', 'Modified By', and 'Checked Out To'. The table contains three entries: 'HowSharePointWorks', 'PDK\_for\_Microsoft\_NET', and 'User Manual'. A left sidebar contains 'Select a View' options (All Documents, Explorer View, EPView) and 'Actions' (Alert me, Export to spreadsheet, Modify settings and columns).

Type	Name	Modified	Modified By	Checked Out To
Document	HowSharePointWorks <span style="color: green;">! NEW</span>	9/10/2005 5:43 PM	Iwan Kostin	
Document	PDK_for_Microsoft_NET <span style="color: green;">! NEW</span>	9/10/2005 5:44 PM	Iwan Kostin	
Document	User Manual	8/18/2005 9:26 AM	Iwan Kostin	

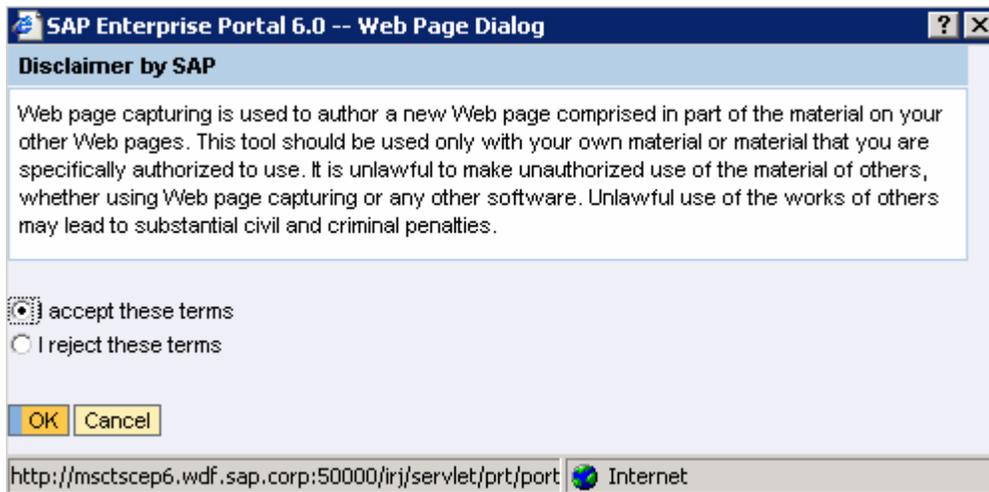
In order to specify which area within the selected Web page has to be displayed in the iView at runtime, change to the *Capture* tab:



Move the mouse cursor over the area you want to select. As you move the mouse, stable HTML elements are highlighted within a peach-colored marquee. The *Selected Element* box indicates the type of element currently selected in the Web page. When the area you want to choose is inside the highlighted marquee, click once. The selected area is marked in a peach color. This is the *primary* capture area.



Once you have selected the content to display in the iView, click **OK** in the portal's built-in browser window. If you read and accepted the onscreen disclaimer, you are returned to the URL iView Editor.



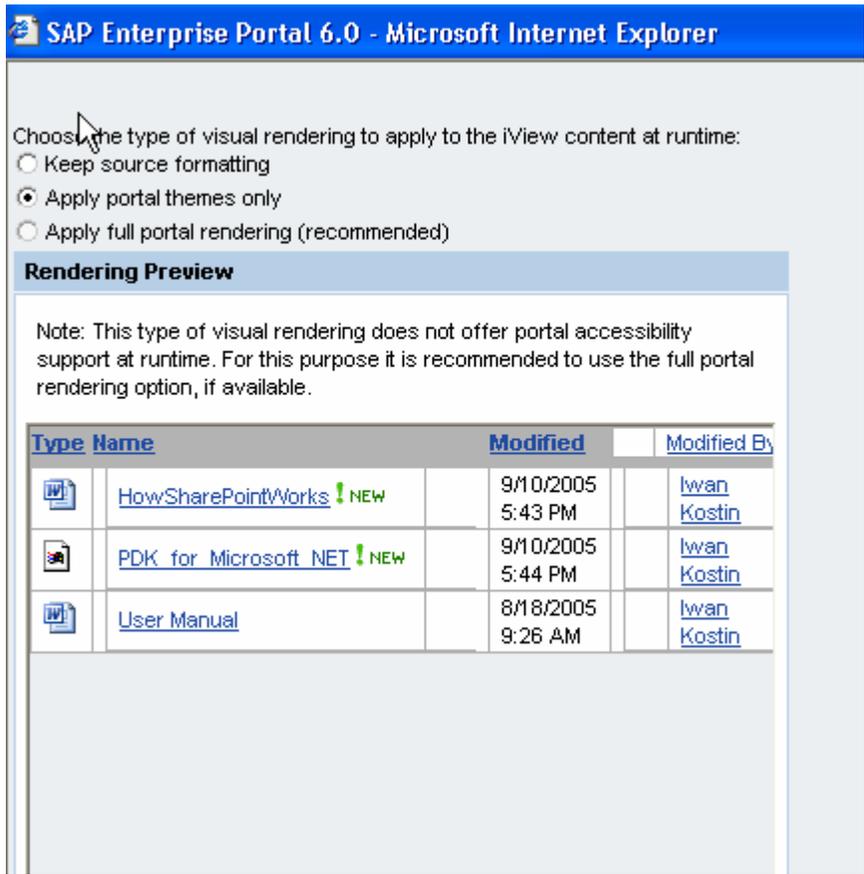
Being in the iView Editor, you can define the way in which content from a Web page will be rendered in a URL iView at runtime.

To modify the rendering of the iView content, click *Edit* next to the *Look and Feel* box. A new window opens in which you define the rendering settings.

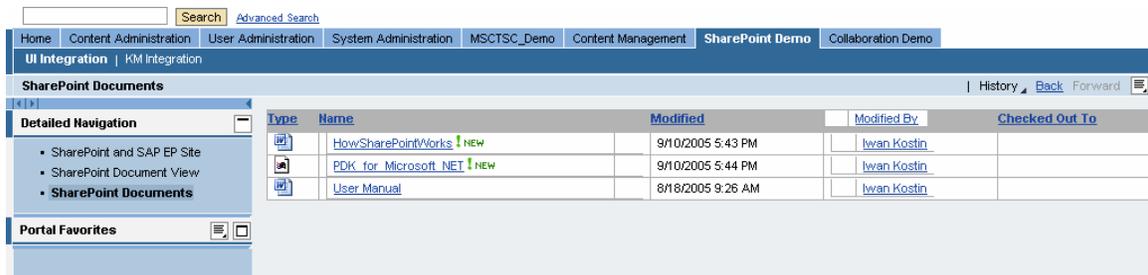
To apply the portal theme currently selected by the end user, choose *Apply portal themes only*. At runtime, the portal substitutes certain styles incorporated by the source Web page, so that the overall result is an approximation of the current portal theme.

**Note:**

The option *Apply full portal rendering* is recommended but it might not always be available depending on the selected area in the Web page to display.



The look and feel of the resulted iView is similar to Enterprise Portal's look and feel:



To partially avoid problems caused by SharePoint displayed within the iView area, you have to set the iView property *Open iView links in the new window* to *Yes*. That enables you to force all hyperlinks displayed in the iView to open in a new browser window when clicked at runtime.

**Note:**

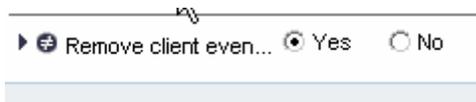
When you enable this option, the performance of the iView is reduced, since the HTML code of the iView content must be filtered and modified before it is sent to the client's browser for display.



After integrating a page area this way, it might happen that you get some JavaScript errors at runtime. To eliminate them, set the iView property *Remove client events from Web site content* to *Yes*. That enables you to remove client events, such as *onClick* and *onMouse*, from the iView before it is displayed at runtime.

**Note:**

Again, when you enable this option, the performance of the iView is reduced.



## How-To: Configuring the SAP WebDAV Repository Manager to integrate SharePoint

### Define an HTTP system in the CM system landscape

An HTTP system is a prerequisite for the configuration and operation of a WebDAV Repository Manager.

To define an HTTP system, choose *System Administration -> System Configuration* in the top-level navigation, and in the detailed navigation *Content Management -> Global Services -> System Landscape Definitions -> Systems -> HTTP System*.

On this page you can duplicate an existing HTTP System or create a new one:

**Edit "msctscsps"**

 Object remains locked until you click OK or Cancel

Description	Default site on MSCTSCSPS
Same User Domain +	<input checked="" type="checkbox"/>
Max Connections +	0
Password +	.....
Re-Enter Password	.....
Server Aliases +	
Server URL *	http://msctscsps.msctsc.sap.corp:1080/
User +	

[Hide Advanced Options](#) (+ denotes advanced options)

### Note:

If the usage of the SSO22KerbMap module is planned, it is important to check the box *Same User Domain* in the configuration of the HTTP system. Activating this option causes that the portal to forward the authentication information stored in a SAP logon ticket to the remote server.

### Hint:

Configuring an HTTP system you specify for the *Server URL* the URL of the remote server. You can either specify the URL targeting WSS virtual server or the URL of a dedicated site collection. It is recommended to use the URL of a virtual server and to specify the same HTTP System in the configuration of different Repository Managers. You can configure appropriate access paths for various repositories by specifying the *System Path* property for the repository manager. *System Path* is an optional suffix for the server URL of the remote system and enables you to use the same system for different paths in the URL namespace.

The advantage of this approach is a better maintainability of user mapping data. Instead of administering user credentials for several systems, you need to do this only once.

You can find a detailed description of configuration parameters in the SAP Online Help [HTTP System](#).

### Create a cache for the WebDAV repository manager

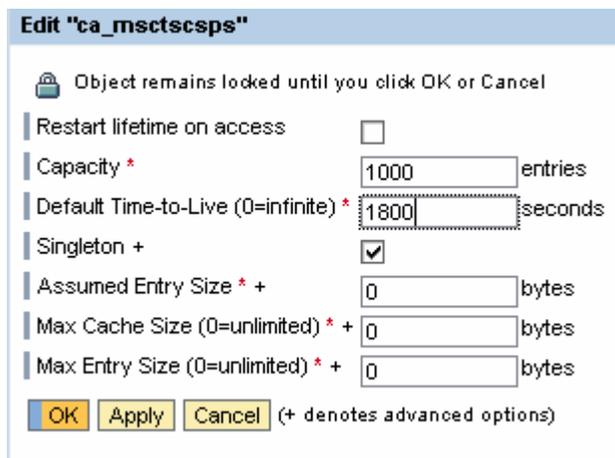
The WebDAV repository uses caching to save roundtrips to the remote server and enhance performance. Caching is done on a per-user basis for systems that require authentication. This has the consequence that each user has its own cache. Therefore the time after which changes are seen by other users depends on the cache timeout.

#### Note:

You should create a separate cache for each WebDAV repository.

To create a cache, choose *Utilities -> Caches* on the *Content Management* iView.

Following configuration is recommended in the SAP Online Help [Components and Their Caches](#).



The screenshot shows a dialog box titled "Edit 'ca\_msctscsps'". At the top, there is a lock icon and the text "Object remains locked until you click OK or Cancel". Below this, several configuration options are listed:

- Restart lifetime on access:
- Capacity \*: 1000 entries
- Default Time-to-Live (0=infinite) \*: 1800 seconds
- Singleton +:
- Assumed Entry Size \* +: 0 bytes
- Max Cache Size (0=unlimited) \* +: 0 bytes
- Max Entry Size (0=unlimited) \* +: 0 bytes

At the bottom, there are three buttons: "OK", "Apply", and "Cancel". A note next to the buttons says "(+ denotes advanced options)".

For detailed information on cache configuration please see the SAP Online Help [Caches](#).

### Configure the WebDAV repository manager

The creation and configuration of a repository manager is the most essential step to integrate an external data source into Knowledge Management. You accomplish the configuration of a WebDAV Repository Manager in the *Content Management* iView if you choose *Repository Managers -> WebDAV Repository*.

The screenshots below show the showcase configuration. To understand the meaning of all listed parameters please read the SAP Online Help [WebDAV Repository Manager](#).

**Edit "msctscsps"**

Object remains locked until you click OK or Cancel

Description

Prefix (must start with /) \*

Active

Hide in Root Folder

System ID (Landscape Service) \*

System Path

Use System Default Proxy Settings

Proxy System ID (Landscape Service)

Proxy Host

Proxy Port

Repository Services

<input type="checkbox"/>	Name
<input checked="" type="checkbox"/>	<a href="#">accessstatistic</a>
<input checked="" type="checkbox"/>	<a href="#">comment</a>
<input checked="" type="checkbox"/>	<a href="#">discussion</a>
<input type="checkbox"/>	<a href="#">discussion_secure</a>
<input type="checkbox"/>	<a href="#">discussion_simple</a>
<input checked="" type="checkbox"/>	<a href="#">feedback</a>
<input checked="" type="checkbox"/>	<a href="#">layout</a>
<input checked="" type="checkbox"/>	<a href="#">personalnote</a>

Page 1 / 3 [Show All Objects](#)

Property Search Manager

Versioning Manager

Security Manager

ACL Manager Cache

Read-only ACLs

Send Events

Cache \*

Cache Timeout +  milliseconds

HTTP Timeout +  milliseconds (0=no timeout)

[Hide Advanced Options](#) (+ denotes advanced options)

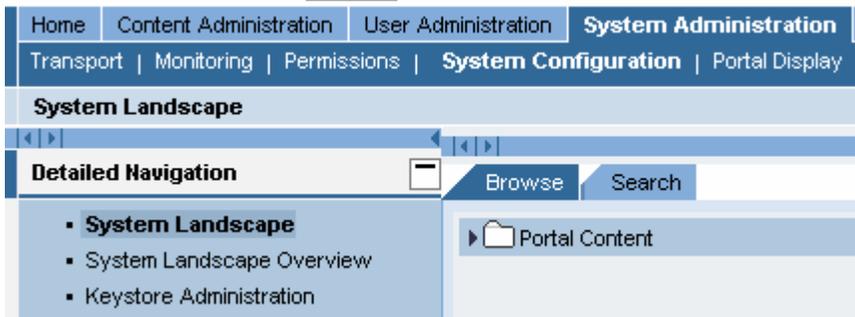
## Create a WebDAV system template

### Note (this step only applies when using user-mapping):

A WebDAV system in the portal system landscape is required for user mapping. If you use the SSO22KerbMap module no user-mapping is required. Since SSO22KerbMap enables the ticket based SSO, there is no need for maintaining user credentials for portal users defined in the user persistence store of Active Directory. However, user mapping should be used for a limited number of internal service users (for more details, see *SSO for internal service users*).

When you install a portal, you also install a number of predefined system templates. Unfortunately, you won't find a template required to create a WebDAV system. Therefore, an additional step is required:

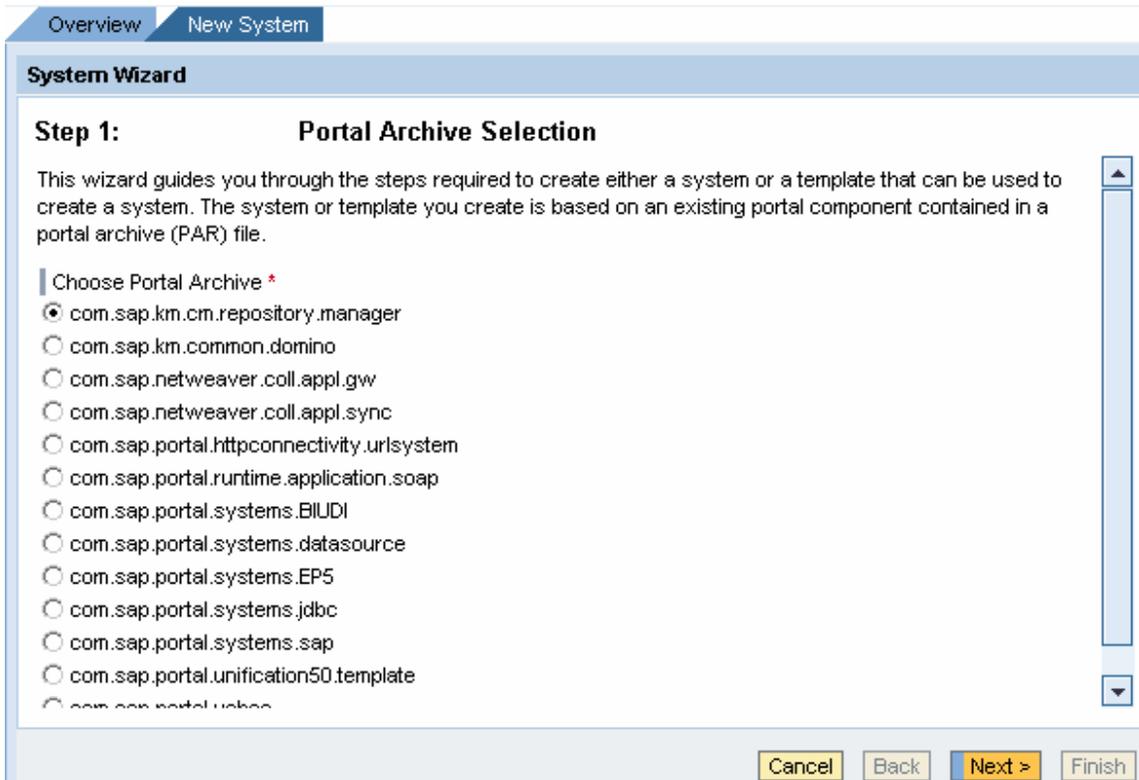
7. Choose *System Administration* -> *System Configuration* in the Top Level Navigation, and in the Detailed Navigation *System Landscape*:



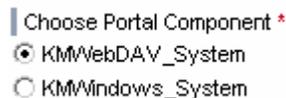
8. In the Portal Content catalog navigate to *Templates* -> *System Landscape Templates*.
9. Right-click the folder *System Landscape Templates* and choose *New From PAR* -> *System* from the context menu:



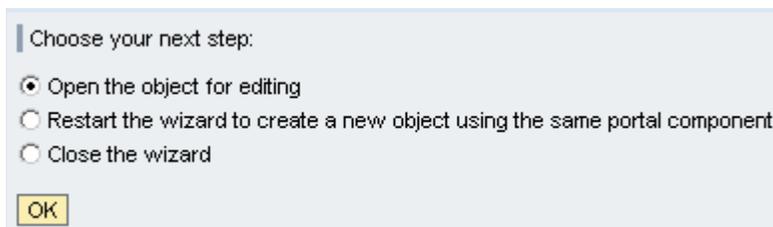
10. In the *Portal Archive Selection* choose the Portal archive *com.sap.km.cm.repository.manager*.



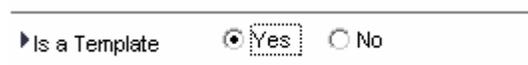
- The selected Portal archive contains two components, choose *KMWeb-DAV\_System*:



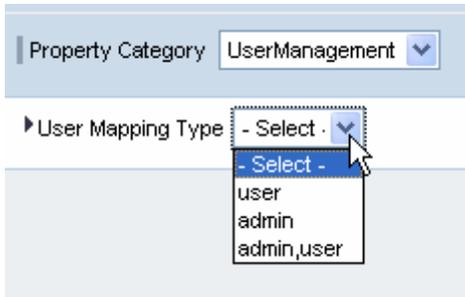
- On the screen *General Properties* specify the system name and the system ID.
- After the confirmation of your entries, open the system object for editing:



- Change the system property *Is a template* to *Yes*:



- In the property category *User Management* choose the user mapping type depending on your requirements and save your changes:

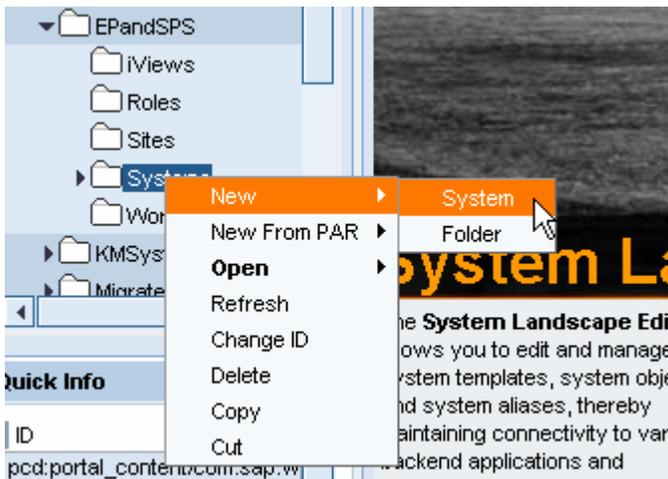


### Create a WebDAV system in the portal system landscape

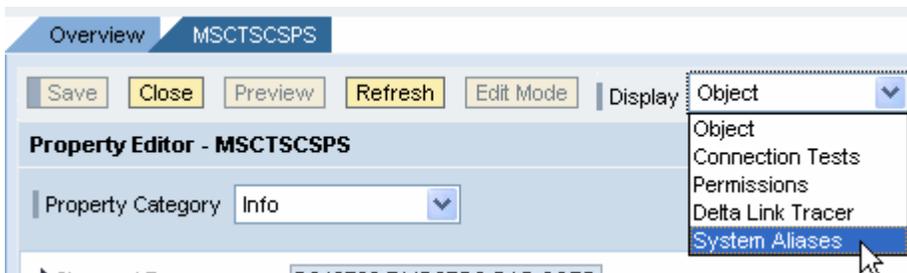
#### Note (this step only applies when using user-mapping):

The procedure for creating a WebDAV system in the portal system landscape is similar to the creation of a WebDAV system template. You use the system template created in the previous step:

1. Choose *System Administration -> System Configuration* in the Top Level Navigation, and in the Detailed Navigation *System Landscape*.
2. Choose any folder in the Portal Content catalog depending on your folder structure in the PCD, right-click on this folder and choose *New -> System* from the context menu:



3. In the template list select the template created in the previous step and click *Next*.
4. On the screen *General Properties* enter a system name and ID and click *Next*.
5. After the confirmation of you entries open the object for editing and select *System Aliases* in the drop down box *Display*.



- Enter an alias for the WebDAV system. Please note that the alias must be the same than the System ID that has been chosen in the CM system landscape definition:

Overview MSCTSCSPS

Save Close Preview Refresh Edit Mode | Display System Aliases

**System Alias Editor**

Create new aliases for this system or delete existing ones. A system alias can be any name y default alias represents the system in other interfaces, such as User Mapping.

Alias  Add

**Defined Aliases**

	Default	Alias
<input checked="" type="radio"/>	✓	msctscsps
<input type="radio"/>		
<input type="radio"/>		
<input type="radio"/>		

- Save your changes and close the editor.

A detailed description on how to create a system in the Portal system landscape can be found in the SAP Online Help [Creating a System in the Portal System Landscape](#).

### Specify user mapping

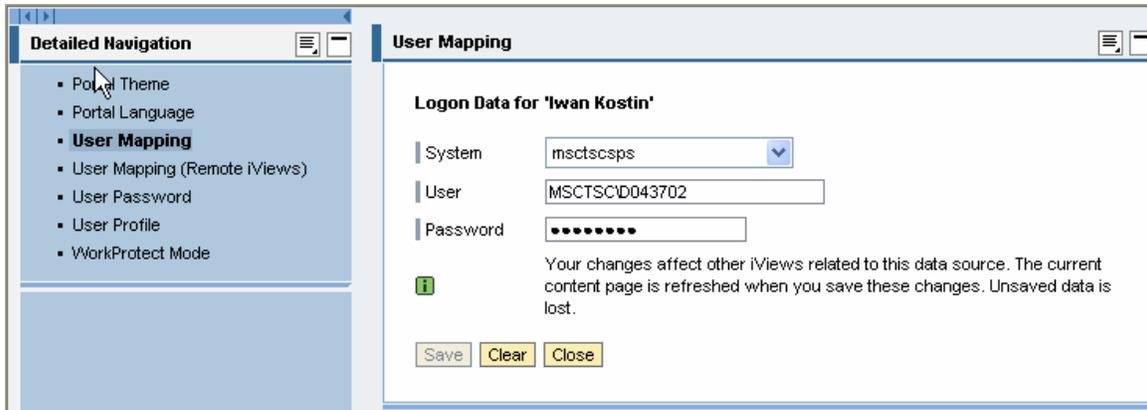
#### Note (this step only applies when using user-mapping):

The repository manager has to provide the credentials for logging on to the remote server for all users that access the remote repository. This information is read from the mapping data of the user who is trying to access the repository.

Depending on how the corresponding system object in the portal system landscape is configured, users can maintain their own mapping data or the mappings have to be made by the user administrator.

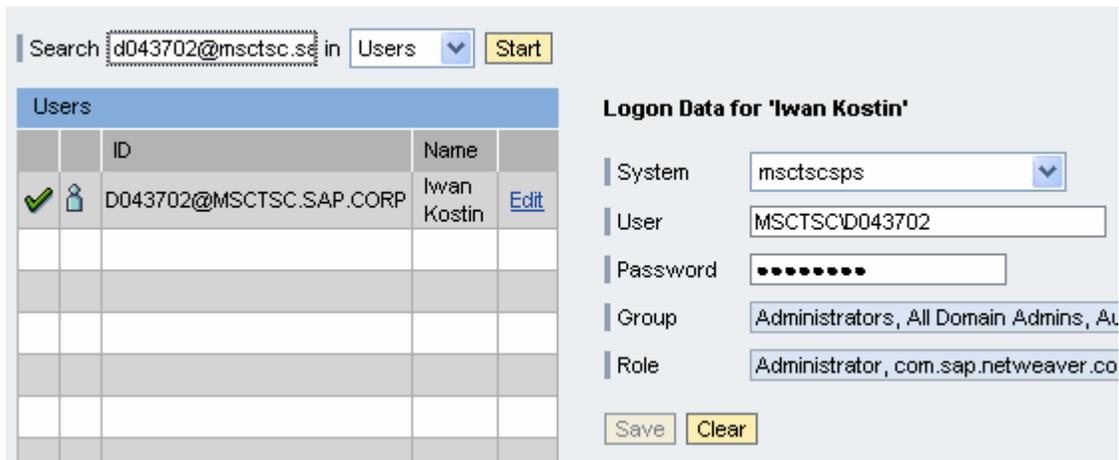
User mapping carried out by a user:

- Choose *Personalize* from the portal header.
- Under *User Mapping*, select the appropriate system, and enter your user and corresponding password for the remote system:



User mapping carried out by a user administrator:

1. In the Top Level Navigation choose *User Administration -> User Mapping*.
2. Call up the settings for the wanted user, and enter the user name to be used and the corresponding password for the remote system:



## How-To: Security

### SSO22KerbMap

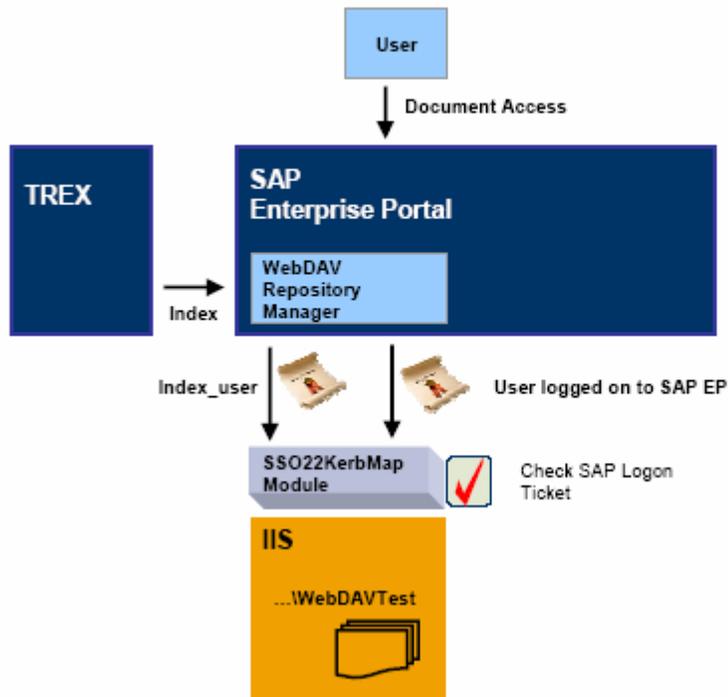
Based on the technology of *constrained delegation using protocol transition* SAP offers the SSO22KerbMap module. SAP's new SSO22KerbMap ISAPI Filter securely identifies the user by the SAP Logon Ticket and requests a constrained Kerberos ticket from Active Directory on behalf of the interactive SAP portal user.

The idea behind the SSO22KerbMap Module is that both SAP Logon Tickets and Kerberos Tickets represent user credentials for their respective security contexts. Constrained delegation using protocol transition allows the use of user credentials represented by a SAP Logon Ticket to obtain a Kerberos ticket which can be used for SSO to MS based backend systems. The SSO22KerbMap Module consists technically out of an ISAPI Filter DLL. The filter allows to obtain Kerberos Tickets on behalf of the user that has been successfully authenticated by the portal and thus has been submitted a valid SAP Logon Ticket.

If the filter DLL is installed as an ISAPI Filter all http requests that are passing the filter will be analyzed. If a SAP Logon Ticket is found in the header of an http requests the portal user id is extracted. The next step performed is the identification of the user in Active Directory based on the information found in the SAP Logon Ticket. Using the configuration file SSO22KerbMap.ini the filter retrieves the information from the parameter *SSO2AccountAttribute* which user attribute in Active Directory is used as portal logon id. It is recommended to use a unique attribute such as the *userPrincipalName* as portal user id. The filter checks for a user in active directory where the value of the attribute matches the value found in the SAP Logon Ticket. If exactly one user is found the ISAPI Filter requests a Kerberos ticket on behalf of that user and adds the Kerberos token to the header.

See also: Collaboration Brief "[Using SAP Logon Tickets for Single Sign on to Microsoft based web applications](#)" on SAP Developer Network

## WebDAV Repository Manager using the SSO22KerbMap Module



For more details on Single Sign on to Microsoft based web applications read following papers:

- [Using SAP Logon Tickets for Single Sign on to Microsoft based web applications](#) (SAP Developers Network)
- [Integration of Windows File Servers into the SAP KM platform using SSO and the WebDAV repository manager](#) (SAP Developers Network)

You can download the SSO22KerbMap from [SAP Service Marketplace](#) (choose in the navigation panel *Download -> Support Packages and Patches -> Entry by Application Group* and then in the work area *Additional Components -> SAPSSOEXT -> SAPSSOEXT -> Windows Server on IA32 32bit*)

### SSO for internal service users

A number of global services and repository services need to perform operations on resources in various repositories. To be able to perform the required operations, the services use service users. The index management service is one of the services provided by CM. It makes documents in CM repositories accessible for search and classification. The service user specified in the service configuration is called by default *index\_service*. It is used when an index is created, and therefore it has to have at least read access to all repositories that are to be indexed. However, when a search is taking place, it is performed with credentials of the user currently logged on to the portal.

In case you implement ticket-based Single Sign-On by means of the SSO22KerbMap module, the setting up SSO for the internal CM service users is handled differently than for other portal users. The reason is that the internal users are only defined in the portal

database. User mapping is not necessary for portal users that are defined in the user persistence store of Active Directory. However, it is an option for the internal service users. The advantage of this approach is that no additional users have to be created in Active Directory and that no additional settings in UME have to be applied to avoid confusion in user management.

### ***Microsoft Enterprise Single Sign-On Service provided by SharePoint Portal Server***

Enterprise Single Sign-On Capabilities in SharePoint environments are provided by SharePoint Portal Server used for intranet scenarios. Windows SharePoint Services do not offer comparable single sign-on support.

Enterprise Single Sign-On (SSO) provides services to store and transmit encrypted user credentials across local and network boundaries, including domain boundaries. The necessary credentials are stored in the Credential database. Middleware applications and custom adapters can leverage SSO to securely store and transmit user credentials across the environment because of a generic solution approach. For end users the access for different applications is transparent and they do not have to remember different credentials for different applications.

A Single Sign-On environment consists of three components:

- A Credential database
- A master secret server
- And one or more Single Sign-On servers

Within the SSO system an administrator defines affiliate applications, representing a remote system such as a back-end system or business application to which users shall connect using SSO services. For each affiliate application there exist multiple user mappings. For example a mapping between an Active Directory Account and a SAP R3 account.

Within the Credential database, being a SQL Server database, all information about the affiliate applications, as well as all the encrypted user/password combinations for these applications are stored.

The master secret is stored on the Master Secret Server which is one of the involved SSO Servers. All the other SSO servers participate in using this master secret. The Credential database and the Master Secret Server have to be unique in a SSO environment.

As a developer it is possible to use the SSO API to fulfill the needs of single sign on within your application.

Remember that in a SharePoint environment it is possible to use the Enterprise Single Sign-On Service only by deploying the SharePoint Portal Services. In order to be able to connect e.g. your own Web Parts to affiliate applications the definition of the affiliate applications as well as the input of the necessary user / password combinations can be done through the SharePoint Portal server administration UI.

## Conclusion

SAP Enterprise Portal and Windows SharePoint Services are powerful products from SAP and Microsoft offering a lot of functionality out of the box. They are also platforms for extensions in customer projects and through 3<sup>rd</sup>-party companies. Due to these functions and capabilities they offer a variety of integration opportunities amongst each other. This Collaboration Brief does address and comment the most common ones but certainly not all of them. Also some of the involved technologies are still evolving. Therefore this paper should be regarded as just a first step towards interoperability between EP and WSS.

## References

- Contact the CTSC about this document: [ctsc@sap.com](mailto:ctsc@sap.com)
- .NET Interoperability on SAP Developer NetWork: <https://www.sdn.sap.com/rdn/dev4areas/netdotnet.sdn?node=linkDnode1-2>
- Wikipedia definition for portlets: <http://en.wikipedia.org/wiki/Portlet>
- SAP Online Help: <http://help.sap.com>