Note: Before using this information and the product it supports, read the information in “Notices” on page v.
## Contents

### Notices
- Trademarks

### Preface
- The team that wrote this Redpaper
- Become a published author
- Comments welcome

### Chapter 1. What is a portal?
- 1.1 What is a portal? .......................... 1
- 1.2 Transforming the business ............... 2
- 1.3 The various flavors of portals .......... 3
- 1.4 Benefits of deploying a portal .......... 4
- 1.5 Portals and their impact on an organization .......... 5
- 1.6 Layout of a typical WebSphere Portal implementation ...... 7
- 1.7 Value-add services ..................... 8
  - 1.7.1 Register on the IBM software support Web site for Passport Advantage .... 8
  - 1.7.2 IBM support .......................... 10
  - 1.7.3 WebSphere Portal Automated Problem Determination Tool ............ 11
  - 1.7.4 Project review ..................... 12

### Chapter 2. Planning a portal
- 2.1 Before you begin ...................... 13
- 2.2 Roles and responsibilities ............. 14
- 2.3 Requirements process .................. 16
  - 2.3.1 Define the goals .................. 20
  - 2.3.2 Project plan ....................... 21
  - 2.3.3 Solution design ................... 21
  - 2.4 Determine and reduce the complexity ...... 22
  - 2.5 Defining non-functional requirements as part of service level agreements .... 23
    - 2.5.1 High availability ................ 23
    - 2.5.2 More about non-functional requirements ..... 25
    - 2.5.3 Calculating the costs .............. 28
  - 2.6 Initial technical considerations ....... 29
    - 2.6.1 Topology planning ............... 29
    - 2.6.2 To cluster or not to cluster ....... 35
    - 2.6.3 Horizontal versus vertical clustering .... 36
    - 2.6.4 Security ......................... 37
    - 2.6.5 Content management .............. 43
    - 2.6.6 IBM Workplace Web Content Management ...... 44
    - 2.6.7 Search ......................... 50
    - 2.6.8 Virtual portals ................ 55

### Chapter 3. Developing a portal
- 3.1 Customization versus advanced personalization .......... 59
- 3.2 Developing portlets .................. 60
  - 3.2.1 APIs and frameworks .......... 60
  - 3.2.2 User interface design .......... 62
  - 3.2.3 Markup generation ............... 63

© Copyright IBM Corp. 2006. All rights reserved.
# WebSphere Portal Best Practices

## Chapter 3. Integration

3.3 Integration .................................................................................. 65
3.3.1 Directory (LDAP) management .............................................. 66
3.3.2 Collaboration components ..................................................... 67
3.3.3 Traditional systems ............................................................... 68

## Chapter 4. Deploying, testing, and maintaining a portal

4.1 Test .......................................................................................... 80
4.1.1 Test processes and environments ......................................... 80
4.1.2 Non-functional tests ............................................................ 83
4.1.3 After going live ................................................................. 88
4.2 Release planning ................................................................. 89
4.3 Deployment ............................................................................. 90
4.3.1 Automating custom code deployment .................................. 91
4.3.2 Staging concepts ............................................................... 91
4.3.3 Clustering and deployment ................................................. 92
4.3.4 Keeping track of the growth ............................................... 95
4.4 Maintenance ........................................................................... 96

## Appendix A. Sample workshop agenda

Appendix B. Sample portal tracking worksheet .......................... 103

## Appendix C. Portlet sourcing

Exercise ....................................................................................... 106
Worksheet ................................................................................... 108

## Appendix D. Solution assurance checklist

Related publications ...................................................................... 131
IBM Redbooks ........................................................................... 131
Other publications ....................................................................... 131
Online resources .......................................................................... 131
How to get IBM Redbooks ......................................................... 136
Help from IBM ........................................................................... 136
Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:
This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.
**Trademarks**

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

<table>
<thead>
<tr>
<th>AIX 5L™</th>
<th>iSeries™</th>
<th>Redbooks™</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIX®</td>
<td>Lotus Notes®</td>
<td>RUP®</td>
</tr>
<tr>
<td>BladeCenter®</td>
<td>Lotus®</td>
<td>Sametime®</td>
</tr>
<tr>
<td>Candle®</td>
<td>Notes®</td>
<td>SurfAid™</td>
</tr>
<tr>
<td>ClearCase®</td>
<td>OmniFind™</td>
<td>Tivoli®</td>
</tr>
<tr>
<td>DB2®</td>
<td>Passport Advantage®</td>
<td>WebSphere®</td>
</tr>
<tr>
<td>developerWorks®</td>
<td>pSeries®</td>
<td>Workplace Web Content</td>
</tr>
<tr>
<td>Domino®</td>
<td>PurifyPlus™</td>
<td>Management™</td>
</tr>
<tr>
<td>@server®</td>
<td>QuickPlace®</td>
<td>Workplace™</td>
</tr>
<tr>
<td>Everyplace®</td>
<td>Rational Unified Process®</td>
<td>xSeries®</td>
</tr>
<tr>
<td>i5/OS®</td>
<td>Rational®</td>
<td>zSeries®</td>
</tr>
<tr>
<td>IBM®</td>
<td>Redbooks (logo)™</td>
<td></td>
</tr>
</tbody>
</table>

The following terms are trademarks of other companies:

- Enterprise JavaBeans, EJB, Java, JavaBeans, JavaScript, JavaServer, JavaServer Pages, JDK, JMX, JSP, JVM, J2EE, Solaris, Sun, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

- Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

- UNIX is a registered trademark of The Open Group in the United States and other countries.

- Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.
Preface

This IBM® Redpaper is designed to provide a road map of information about how to best plan and ensure a successful deployment of IBM WebSphere Portal into an organization.

In this Redpaper, we provide you with an understanding of the WebSphere Portal technology and discuss common goals for customer portal projects. In addition, we discuss initial planning considerations and how to plan your architecture for deployment.

Best practices for a successful portal deployment requires a systematic process. Therefore, we address the initial questions that are asked to get you started and keep you on the right path. You will find that this best practices guide also serves as your systems assurance lead, ensuring that IBM products are planned, shipped, installed, and tested in such a way that clients derive maximum satisfaction with minimal disruption.

This document is designed for the following audience: IBM employees, IBM Business Partners, and clients.

The team that wrote this Redpaper

Residency team (left to right): Bernhard Stimpfle, Jeanette Coury, and Rufus Credle

This Redpaper was produced by a team of specialists from around the world working at the International Technical Support Organization, Raleigh Center.

Rufus Credle is a Certified Consulting I/T Specialist at the International Technical Support Organization, Raleigh Center. He conducts residencies and develops IBM Redbooks™ about network operating systems, ERP solutions, voice technology, high availability and clustering solutions, Web application servers, pervasive computing, and IBM and OEM e-business applications, all running IBM @server® xSeries® and IBM @server BladeCenter® systems. Rufus’s various positions during his IBM career have included assignments as an
administration manager, systems engineering, sales and marketing, and IT services. He holds a B.S. degree in business management from Saint Augustine’s College. Rufus has been employed at IBM for 25 years.

Jeanette Coury is a Consulting IT Specialist working in IBM Sales and Distribution, Software Sales for Lotus® in the Mid-Atlantic Business Unit. She has more than 25 years of experience in the IT industry and holds a Bachelor of Science degree in Business Administration from Towson University. Jeanette is an IBM Certified System Administrator for WebSphere Portal, WebSphere® Application Server, and Lotus Notes® and Domino®. Jeanette joined IBM in 2000 as a Presales IT Specialist working with Integrated and Aligned Accounts, as well as SMB customers. She started as a technical expert for Lotus Domino and the Domino family products and soon specialized in WebSphere Portal, knowledge management, and Web content management.

Bernhard Stimpfle is a Portal and Pervasive Solutions Architect for the IBM Lab-based services in Boeblingen, Germany. He reviews architectures, has been Lead Architect on WebSphere Portal projects, and supports major customers on-site in critical situations. Bernhard has worked for more than five years with IBM. He has spent a total of 10 years in the IT industry, working for Daimler-Chrysler Aerospace and managing his own business. His area of expertise include WebSphere Portal, pervasive computing solutions, UNIX®, Java™ 2 Platform, Enterprise Edition (J2EE™), and general multi-tier architectures. He is a Red Hat Certified Engineer (RHCE) and holds a Diplom-Ingenieur degree in Computer Science from Berufsakademie Ravensburg, Germany.

Thanks to the following people for their contributions to this project:

Emma Jacobs, Jeanne Tucker, Tamikia Barrows, Margaret Ticknor
International Technical Support Organization, Raleigh and Almaden Center

Tony Higham, Executive IT Architect, Portal and Content Management
IBM Atlanta

David Kruse, Technical Business Leader for Business Generation, Americas TechWorks
IBM Cleveland

Brian Meyer, IBM On Demand Workplace™ Framework Architecture, Intranet Search Portal
Day In The Life and Architect
IBM Boulder

Dave Minear, IBM Distinguished Engineer, Certified IT Architect
IBM San Francisco

Stefan Liesche, Workplace and Portal Foundation Lead Architect
IBM Germany

Michael Menze, WebSphere Portal Performance Architect
IBM Germany

Steffen Uhlig, WebSphere Portal Solutions Architect
IBM Germany

Thomas Stober, WebSphere Portal Release Architect
IBM Germany

Ralf Duerig, IT Architect
IBM Germany

Ekkehard Schulz, Executive IT Architect, IBM Software Group Services
IBM Germany
Tom Alcott, Consulting IT Specialist, Worldwide Technical Sales
IBM Costa Mesa

Bill Hines, Senior Certified Consulting IT Specialist, IBM Software Group
IBM Mechanicsburg

Skyler Thomas, Senior Technical Staff Member, SOA Architect, IBM Software Services
IBM Raleigh

Wolfgang Raestrup, IBM Software Group Services
IBM Germany

Andreas Prokoph, WebSphere Portal Lead Architect Search Technologies
IBM Germany

Marco Seifried, Solutions Architect, IBM Software Group
IBM United Kingdom

Anthony Bernal, Senior Consulting IT Specialist, IBM Software Services for Lotus, Solutions Architect, IBM Software Group
IBM Houston

Ken Polleck, Senior Practice Manager, IBM Software Services for Lotus
IBM Raleigh

**Become a published author**

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll team with IBM technical professionals, Business Partners and/or customers.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you'll develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at: [ibm.com/redbooks/residencies.html](http://ibm.com/redbooks/residencies.html)

**Comments welcome**

Your comments are important to us!

We want our papers to be as helpful as possible. Send us your comments about this Redpaper or other Redbooks in one of the following ways:

- Use the online **Contact us** review Redbook form found at: [ibm.com/redbooks](http://ibm.com/redbooks)
- Send your comments in an e-mail to: redbook@us.ibm.com
- Mail your comments to:
  IBM Corporation, International Technical Support Organization
  Dept. HQ7 Building 662
  P.O. Box 12195
  Research Triangle Park, NC 27709-2195
What is a portal?

This IBM Redpaper is designed to be a road map for a successful IBM WebSphere Portal project implementation within your organization. The target audience is customers or IBM Business Partners that are new to the technology. This paper is your guidebook to the WebSphere Portal implementation process. There are numerous published papers, books, and technical articles about WebSphere Portal server. We condensed many of those topics into this Redpaper and provided the links to those resources for your review. This guide is a good starting place for a new WebSphere Portal project. It is designed to help you learn the critical paths to consider and the tasks with the most risks. We outline a staged implementation approach and share lessons learned from hundreds of WebSphere Portal implementation projects.
1.1 What is a portal?

A portal offers a single point of personalized, unified access to applications, content, processes, and people. A portal delivers integrated content and applications, plus offers a unified, collaborative workplace. A portal also provides other valuable functions such as security, search, and workflow. A portal is an open, standards-based framework supporting a wide array of options across databases, directories, platforms, and security. Portals are the next-generation desktop, delivering e-business applications over the Web to many different client devices. Portals are designed to meet the needs of all enterprises, from small and medium businesses to the largest enterprises that demand the most scalable, secure, and robust infrastructure.

“A complete portal solution should provide users with convenient access to everything they need to get their tasks done anytime, anywhere, in a secured manner.”

Stefen Liesche, IBM Portal Architect

A consistent, integrated user experience is achieved by portals that do not only aggregate components into a single view, but in addition allow integration of these components within the context. This is often called integration on the glass, because all the applications are integrated in context by the portal into one single window on the monitor of the portal end user. This is a very powerful concept that in today’s world of widely fractured IT infrastructures enables the delivery of consistent and integrated views on multiple IT services. Integration on the glass improves the user experience and productivity of the IT user; instead of dealing with different IT systems with potential different user interfaces, integration on the glass provides a single, consistent view.

In addition to contextual integration capabilities, portals can provide rich programming frameworks for building user interfaces for component-oriented applications in service-oriented architectures. Service-oriented architecture (SOA) is an approach for building distributed systems that deliver application functionality as services to either end-user applications or other services. SOA provides means to integrate and manage these different services. For more information, refer to the following Web page:

http://www.ibm.com/SOA

Portals provide first-class user interface (UI) support in service-oriented architectures. Portlets, their basic building block, let developers focus on unique aspects of their application, while the middleware handles common functions for life cycle, per-user customization, aggregation, and integration with other components. In addition, portals might provide valuable service functions such as security, search, collaboration, and workflow. Portals provide the ability to aggregate and integrate the UI in a similar way SOA run times can combine and integrate services. Component UIs are aggregated into larger, higher value UIs, giving users a single view of IT services with a single UI to master. Applications originally designed separately can be integrated (aggregation and context) together to enable new function. The portal model allows for improved agility for on-demand businesses. Portal administrators become application integrators who create new applications for their users without programming: by defining new pages, adding portlets to them, connecting the portlets together in context, and setting entitlements. With portal technologies, end users can become their own application assemblers by customizing their portal-based workspaces.

There are many reasons why a portal would benefit your organization, for example:

- Control information glut
- Improve cycle times
- Empower knowledge workers
Chapter 1. What is a portal?

- Reduce it complexity
- Enhance partner and supplier communication
- Streamline processes

WebSphere Portal (Portal for short) supports multiple industry portals and various communities within a company. Portal consists of four basic services: Framework, Integration, Content, and Collaboration. Figure 1-1 illustrates the IBM WebSphere Portal framework.

**IBM WebSphere Portal:**
- Helps build scalable and reliable portals that help improve employee productivity and increase customer loyalty.
- Delivers a single point of personalized interaction with applications, content, processes, and people.
- Integrates business processes and portal users through orchestrated workflow.
- Features IBM Workplace Web Content Management for keeping your portal up-to-date, accurate, and in control.
- Provides powerful collaboration capabilities such as instant messaging, team workplaces, people finder, and e-meetings.
- Enables quick portal integration with back-end systems through portlet builders and open standards.

![Figure 1-1 IBM WebSphere Portal framework](image)

### 1.2 Transforming the business

Today, titles, positions, and personal network are the keys to unlocking the doors to get work done. Getting the right information in a timely fashion takes individual effort, takes a lot of time, causes a lot of stress, and is not always the best approach. Finding the right people who have answers or who can help requires an extensive amount of active work. A manager is expected to do more, is equipped to do less, and has less people and budget. The intranet offers a wealth of information but takes an incredible effort to mine. A typical intranet today has hundreds if not thousands of pages.

"An enterprise whose business processes—integrated end-to-end across the company and with key partners, suppliers and customers—can respond with speed to any customer demand, market opportunity or external threat."

Sam Palmisano, IBM Chairman and CEO
1.3 The various flavors of portals

WebSphere Portal is a key component of the IBM strategy for On Demand Business.

A WebSphere Portal On Demand Workplace:

- Provides *one place* with personalized access to your resources.
- Integrates content, learning, expertise, collaboration, and business applications.
- Enables increased productivity through role-based delivery of resources.
- Eliminates development deployment costs through reuse.

Roles will open the doors to content and people. The right level of information will come to a person based on their role. Finding experts and communities to help will be driven by experiences. A manager is able to do what is expected because of the just-in-time instruction and training. The intranet is an incredible asset that helps deliver what every person needs.

Portals will be the mechanism to support this next on demand computing era. The vision is to move from unit portals and stand-alone applications to a single workplace available to support roles.

Portal solutions can be designed to fit the needs of a particular target audience or business segment. There are business-to-business (B2B), business-to-employee (B2E), business-to-consumer (B2C), and collaboration portals.

A B2B portal solution facilitates quick communication between company employees, company business processes, and suppliers. This type of portal is designed to result in shorter order-to-delivery cycles, an increase in productivity, and a reduction in bottlenecks.

A B2E portal is designed to increase productivity and reduce costs by providing a single, consistent interface. The main design goals are typically to integrate existing applications and increase communications while reducing complexity. Faster access to people, processes, and information and improved communications are added benefits.

A design goal of a B2C portal solution is to support better customer service and increase revenue. This type of portal helps enable a business to be more responsive to customers and provide accurate and up-to-date information while increasing the customer's loyalty to the business.

The collaboration portal is designed to facilitate collaboration, communication, and human interaction. The goal of the collaboration portal is to catapult productivity to higher levels and enable geographically dispersed teams to work as though they were all in the same location. This type of portal provides content management services, the mining and organization of related information, along with collaborative services that allow users to chat, e-mail, share calendars, and define user communities. Collaborative portals are typically internal corporate portal installations.

IBM began using WebSphere Portal in April of 2004 when they converted from their Web-based intranet to a portal-based intranet. The intranet site provides IBM employees with three major areas of information: Home, Work, and Career and Life, all of which contain portlets that can be modified to meet the needs of each individual employee. Employees can chose the type of information they would like to see and they can search for other employees within the company, and they can do this in several different ways.

To help the company and its clients grow their businesses, IBM needed a more powerful platform to tap into its greatest competitive advantage—the collective knowledge, experience, and expertise more than 300,000 employees.
Business benefits include:
- Faster accessibility to people, processes, and information
- $680 million annual cost savings
- Employee productivity increased by 1-3 hours/month
- 90% of U.S. IBM personnel enroll online for benefits
- 80% reduction in average expense processing costs

Technical benefits include almost 1 million visits daily through a single, unified interface, accessing:
- Employee directory
- Human resources online services
- Travel services and expense management
- Manager's corner
- Learning center

1.4 Benefits of deploying a portal

Customers choosing one of the WebSphere Portal offerings can realize tangible business and technical benefits:
- Revenue benefits, as a result of tighter relationships with customers or partners, workforce productivity, innovation, and reduced cycle times
- Operational cost reduction, as a result of operational efficiency, better information flow and knowledge, and consistent infrastructure
- Increased employee productivity and improved decision making because of access to more relevant information, and a single access point to applications and collaboration tools
- Better security and single sign-on, resulting in fewer passwords to administer and better user experience
- Reduced training costs, resulting from common presentation and a consistent user interface
- Unification of applications, giving them a longer useful life, with new ways of accessing them through the desktop and pervasive devices

1.5 Portals and their impact on an organization

An IBM On Demand Workplace provides a platform for integrated service delivery across corporate functions and business units. Therefore, there is not one owner of all business applications, or a natural owner within most organizational structures.

Governance

IBM On Demand Workplace governance will establish a structure of shared ownership and accountability for strategy, development, and operations.

According to a Forrester Research Report, Making Enterprise Portals Pay, August 2001, 71% of the respondents stated that the biggest implementation challenge is managing organizational issues for an enterprise portal.

Along side the technology, content, and services, successful portal implementations include governance.
Therefore, it is important to establish a committee who will govern the design of the portal. Design includes, but is not limited to, security, navigation, and system integration. This might be very difficult to implement if your corporate culture has previously allowed ad-hoc Web page and content publishing. The governance committee will work along side your architects to prioritize and validate business requirements and to review solutions for feasibility.

For example, a governance committee might created standards for:

- Content management
- Security/authentication
- Data warehousing
- Profile management
- Search and taxonomy
- Enterprise brand (look, feel, navigation)
- Metrics/scorecard

**Important:** Governance is needed not only during the initial implementation but also during ongoing expansion.

**Note:** It is paramount to include executive sponsorship in your governance model.

Governance spans your portal strategy, decision-making, and standards. Governance not only addresses internal requirements, such as a corporate-wide look and feel to the portal, but also external requirements, such as portal integration with existing applications.

Other members typically included in the governance framework include corporate communications, IT services, application development, and human resources.

Within the IBM deployment of an On Demand Workplace global Web architecture (GWA), standards have been applied to ensure consistency in content and design processes, yielding a 20% reduction in the cost to deploy Web applications.

All portal pages have the following characteristics: a theme, skins, portlets pages and navigation, and content. Figure 1-2 on page 7 shows the layout of a sample portal page.
Themes and skins, in association with the portal engine and navigation tag libraries, are responsible for the page layout, look and feel, and navigation. The portal servlet knows nothing about what the interaction model should look like or how it behaves; it simply dispatches the theme and skin JavaServer™ Pages™ (JSP™) that do all of the rendering work.

Portlets
Portlets provide the means of delivering applications and content on the portal. The term portlet refers to a small reusable program that can be placed on the portal page to perform a specific function, such as retrieve and display a piece of information.

Themes
Themes provide the navigation, appearance, and layout of the portal, including colors, fonts, and images outside of the portlet content area (Home screen).
Screens
Screens fill the area of the portal that typically displays portlets (Home screen), but can also display other content in its place, for example, a login form or error message. Screens are selected from navigation icons in the theme.

Skins
Skins represent the border rendering around components, such as row containers, column containers, or portlets. Skins are installed independently from themes. However, the administrator can set a default skin for a theme.

Note: The starting place for building the portal page is Default.jsp in the /themes directory. The screen and skin are called by the corresponding <portal:screenRender/> and <portal:pageRender/> tags from the engine tag library.

Theme and skin JSPs, along with the “screen” JSPs responsible for such things as error display, login, and logout, are stored under the main portal EAR (wps.ear) as HTTP addressable entities. Themes can be assigned for top-level pages only; all sub-pages inherit the theme from the top-level. Skins can be assigned on a per-portlet basis, and control how the portlet should be decorated as it is rendered, such as border and skin button styles.

Themes and skins are, therefore, the natural extension point for customers to add their custom navigation or portal-wide business logic (hidden pages, global variable space, and so on). The problem with this is that this makes themes and skins very difficult to migrate, because of the custom Java and JSP code. Migration is also complicated because the portal model tag libraries, used as the basis for navigation tree rendering, are not yet solidified and public.

Tip: WebSphere Portal ships updated versions of its themes with every version, so customers must analyze the differences between the old and new themes to understand what must be migrated forward and how. Documentation of the model API and tag libraries is available from the WebSphere Portal support site to aid in this process.

1.7 Value-add services

When you purchase IBM software, you get more with the product than just the installation CDs. You are now entitled to a number of value-add services, programs, and processes. One of the most important is Software Maintenance. This provides you with upgrade protection and technical support and is included with each IBM distributed software license (including IBM WebSphere, DB2® Information Management, Lotus, Rational®, and Tivoli® software). It is sold through the IBM Passport Advantage® and Passport Advantage Express programs. Each license includes Software Maintenance coverage and entitles you to access the latest versions and releases of software and technical support for your IT staff.

You can receive media delivery of new releases or versions of enrolled software licenses or you can download product updates as soon as they become available. To access this feature through the password-protected Passport Advantage Online, visit:

http://www.ibm.com/software/passportadvantage

1.7.1 Register on the IBM software support Web site for Passport Advantage

Passport Advantage is the IBM comprehensive software licensing and software maintenance program. Maintenance includes product upgrades and technical support.
Use the following the process to register for the first time. Note that if you encounter any problems after you have registered, do not try to re-register, because this will block your access and an IBM technician will need to remove the second registration before you can access the account.

If you get an “Error Unauthorized” message or other error message, it might be a problem with the data records that we have. Send an e-mail to the Electronic Service Request (ESR) help desk (mailto:esrhelpdesk@us.ibm.com) and we can help solve your problem. Include your IBM customer number and your user name and password.

Complete the following common steps for all users:

1. Create your new IBM registration user ID and password:
   a. Access the IBM software support problem submission page at:
   b. Select the ESR link (it has a yellow key icon next to it).
      The IBM software support sign-in page opens.
   c. If you already have an IBM registration user ID and password (from accessing the Passport Advantage customer site), enter your IBM registration user ID and password and click Submit.
   d. If you do not already have an IBM registration user ID and password:
      i. If you do not have an ID, select the Register link.
      ii. The IBM registration form opens. Complete the form. Required information is indicated with an orange asterisk (*). Click Continue at the bottom of the form.
      iii. The registration confirmation page opens. (If an error message appears, follow the instructions on the page to resolve the problem.) Click Continue.
      iv. The sign-in page opens again. Enter the IBM registration user ID and password you just created, and click Submit.
      v. The My Maintenance Agreements page opens.

2. Associate your new IBM registration user ID and password with your IBM customer number:
   a. In the “IBM customer numbers” section, enter your IBM customer number to identify your Passport Advantage Agreement.
   b. In the “If you are an authorized caller” section, enter your First name, Last name, and E-mail address.
   c. Take one of the following steps, depending on your role:
      • If you are the Primary Site Technical Contact, enter your IBM customer number, First name, Last name, and E-mail address exactly as they appear on the Passport Advantage Welcome letter received by the Primary Contact of the Passport Advantage site or the Remote Technical Software Support Information e-mail/letter received by the Site Technical Contact of the Passport Advantage site.
      • If you are a Secondary Site Technical Contact or an Authorized Caller, enter your IBM customer number, First name, Last name, and E-mail address exactly as they appeared in the Authorized Caller e-mail received from your Primary or Secondary Site Technical Contact.

3. Complete your registration for the IBM software support Web site:
   a. At the bottom of the page, click Submit.
b. If all of the information you entered is correct, you will see a confirmation page. To the right of each section, you should see a green circle and some text indicating that the information has been validated. Click Continue at the bottom of the page. This takes you to ESR, where you can submit and track problems.

**Note:** If you see a red circle to the right of the “IBM customer number” section or the “If you are an authorized caller” section, send an e-mail to mailto:esrhelpdesk@us.ibm.com to report the problem. In your message, include all of the information you entered in the My Maintenance Agreements form and your IBM registration user name (which you created on the IBM registration page).

**Note:** If you see a red circle to the right of the “PartnerWorld for Software” label in the “Other IBM support services” section, do not worry. This does not affect your ability to submit and track problems in ESR.

The following steps describe the specific steps for the primary site technical contacts:

- **Secondary Site Technical Contact/Authorized Caller Management steps:**
  a. Access the IBM software support Web site at:
     
     http://www.ibm.com/software/support/
  b. Select the **Submit & Track Problems** link (left side of window).
  c. On the Problem Submission page, select the **ESR** link (it has a yellow key icon next to it) next to the sentence “a customer with an IBM Passport Advantage or Tivoli support contract.”

- **If you are a customer with an IBM Passport Advantage or Tivoli support contract:**
  a. On the sign-in page, enter your IBM registration user ID and password, and click **Go**.
  b. Select the blue circle under the Edit Info column.
  c. On next page, you can either add or update the Secondary Site Technical Contacts and Authorized Callers. (Secondary Site Technical Contacts have the same management ability as the Primary Site Technical Contact, except they cannot update the Primary Site Technical Contact's information.)

When you add a Secondary Site Technical Contact or an Authorized Caller, registration instructions are sent to these through e-mail. The e-mail also includes the exact details needed to associate an IBM registration user ID and password to an IBM customer number (for example, IBM customer number, first name, last name, and e-mail address of the Secondary Site Technical Contact/Authorized Caller).

### 1.7.2 IBM support

The IBM software support organization is a global network of centers with expertise across our broad product portfolio. The organization is made up of teams of individuals that work together to provide you with the responsive software support that you require. Our worldwide centers are structured to provide you with local language access in most major countries and with the skills to help you identify the source of your problem among the products for which you have purchased support.

All IBM customers are entitled to take advantage of the self-help services, available at:

http://www.ibm.com/software/support
IBM offers a vast range of online service offerings designed to augment and enhance the value of your IT operation. With these resources and tools, our self-help software support Internet site will meet many of your support needs. When you have an issue, make this your first response. You can search closed Problem Management Records (PMRs), consult white papers and redbooks, and consult forums.

When you open a PMR, you need to define the problem, gather background information such as the level of software you are running, gather diagnostic information (if appropriate), and determine the business impact. To determine the business impact, refer to the chart in Table 1-1.

### Table 1-1 Business impact

<table>
<thead>
<tr>
<th>Severity level</th>
<th>Severity definitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Critical impact/system down: Business-critical software component is inoperable or critical interface has failed. This indicates that you are unable to use the program resulting in a critical impact on operations. This condition requires an immediate solution.</td>
</tr>
<tr>
<td>2</td>
<td>Significant impact: A software component is severely restricted in its use, causing significant business impact. This indicates that the program is usable but is severely limited.</td>
</tr>
<tr>
<td>3</td>
<td>Moderate impact: A noncritical software component is malfunctioning, causing moderate business impact. This indicates that the program is usable with less significant features.</td>
</tr>
<tr>
<td>4</td>
<td>Minimal impact: A noncritical software component is malfunctioning, causing minimal impact, or a nontechnical request is made.</td>
</tr>
</tbody>
</table>

Use the following steps to submit or track a problem (PMR) if you are a customer with an IBM Passport Advantage or Tivoli support contract:

1. Access the IBM software support Web site at:
   http://www.ibm.com/software/support/
2. Select the **Submit & Track Problems** link (left side of the window).
3. On the Problem Submission page, select the **ESR** link (it has a yellow key icon next to it) next to the sentence “a customer with an IBM Passport Advantage or Tivoli support contract.”
4. On the sign-in page, enter your IBM registration user ID and password, and click **Go**.
   You are now at Electronic Service Request and Authorized Caller Administration page.
5. Click the **IBM customer number** link, which opens the IBM Customer Support Welcome Page, where you can report a new problem and work on existing problems.

**We strongly encourage you to review the IBM Software Support Handbook, available at:**
http://techsupport.services.ibm.com/guides/handbook.html

This guide introduces you to the people in IBM support and explains the policies and procedures. It also provides you with instructions about how to phone into support if you do not have access to the Internet.

### 1.7.3 WebSphere Portal Automated Problem Determination Tool

This tool provides automatic data collection and symptom analysis support for WebSphere Portal problem determination. This tool helps reduce the amount of time it takes to reproduce
a problem because tracing levels are set for you. It also reduces the effort required to send the appropriate log information to IBM support. To download the tool, go to:


1.7.4 Project review

Because this is a best practice guide about how to successfully implement a WebSphere Portal-based solution, you might assume that you are already in good shape now. However, there is more to do than just read this paper and the related resources. Throughout the book, we use the expression “it depends,” because not all questions regarding the usually complex WebSphere Portal projects can be answered with a simple yes or no.

There will be cases where it is a great advantage to have somebody on your team that is able to answer questions with the background of experience with many projects, and there will be situations where you might just need bold decisions.

For the first part, IBM is able to help you. In addition to the consultants from many IBM Business Partners, IBM Global Services, and IBM Business Consulting Services, there are also consultants that do not belong directly to IBM Software Group. All of these consultants work closely together. In addition, you can get a project review from an experienced Software Group Services or Lab-Based Services architect.

Many issues and problems that appear in the very last phase of the project could have been detected and prevented right at its beginning. Even projects with tight budgets have the chance to go for a one-day or two-day remote workshop. Talking with experienced architects about your topics might also give your staff more confidence in their decisions and they might bring up other questions that were not considered.

For the second part, we need you.
Planning a portal

If you are responsible for defining and delivering your organization’s solution offerings, you undoubtedly know the importance of planning. A successful WebSphere Portal deployment requires attention to several key details up front. This chapter discusses the vital role of planning in a portal project.
2.1 Before you begin

Thorough planning for a WebSphere Portal (Portal for short) implementation is the key to success. Portals, if done correctly, will eventually be the front end for everything in your organization, from news to internal applications, documents, search, calendar, and mail.

Do not plan to build an all-encompassing super portal as your first project. Do not start the project by loading the CDs into the drive. Careful planning will reward you in the end. Start with a small project and build from the base components first. You can easily install added functionality later.

Know that WebSphere Portal will test your organizational efficiency. Most customers have different IT staff supporting the various back-end applications. This staff will need to work together on a portal project. Cooperation and coordination will be key to a successful outcome. Learn from the experiences of hundreds of customers. The following sample portal project breakdown lists the high-level breakdown and the reasonable time lines you should expect:

- Requirements planning (one week to ...)
  Collecting and summarizing business requirements. In addition, take advantage of some IBM processes at this stage, architecture and design workshops, and the Business Value Assessment for Portal. We cover these in more depth in 2.3.1, “Define the goals” on page 20.

- Project startup (20 to 30 days)
  Identify the team players and use this time to build the skills for both development and infrastructure support. Set expectations and time lines and create the project plan. See 2.3.2, “Project plan” on page 21 for more details.

- Solution definition (20 to 30 days)
  This is where the business requirements are translated to a technical architecture. Designs are documented and a project road map is refined. You might also request an IBM Solution Assurance Review at this stage. See 2.3.3, “Solution design” on page 21.

- Project standards (three days)
  Identify your change management process. Develop your documentation methodology and testing procedures.

- Environment setup (six months elapsed time)
  Get an IBM Techline sizing and procure your hardware and software. Install your environments and begin base-line stress testing. Refer to 4.1.2, “Non-functional tests” on page 83 for more details.

- Pilot release (one to two months)
  Select and design portlets. Document and run your use cases. Perform some preliminary stress testing. Refer to Appendix C, “Portlet sourcing” on page 105.

- Production release (length of pilot plus three months)
  Select more pilot projects and build onto the pilot. Add more functionality like Web content management or enabling security. Perform more stress testing and test fail-over procedures.

- Project close (three days)
  A postmortem meeting is a good way for the project manager to record viable information that might prove useful for the next portal endeavor. Ensure that ongoing administration is in place and develop any follow-on plans.
This is just an example, and your situation might differ in one or another point. It is meant to give a rough outline (we see many similar outlines being used). You might, for example, want to do more pilot release iterations if you have the requirement to start with a full bucket of features in your production release.

The main lesson that you should learn here is not to underestimate the complexity of a portal solution. That is why you require a pilot release and that is also why your data center staff requires some time to handle the new situation: It is often less the complexity of the portal itself, but the many components and people in your IT infrastructure that are now supposed to work together in a way they have never done so far.

**Portal project length**

Review the time line of a portal project in Figure 2-1. This was created based on statistics gathered by the IBM Software Group Services Lotus. Remember this information when setting expectations within your organization. Do not yield to pressure from project sponsors to deliver sooner. You will find yourself negotiating with them near the end of your project to drop needed functionality in order to meet the deadline. This might lead to scrutiny surrounding your project management skills, the technical skills on the team, and even back to the original purchase decision to buy WebSphere Portal.

**Tip:** Expect a four month minimum for the first pilot release.

![Portal Project Length](image)

**Figure 2-1  Time line of a portal project**

Basically, the first three-to-four months is considered your pilot stage, while you spend time connecting to established systems, understanding the functionality, developing a limited deployment strategy, and building portlets. Other moments during the time line consist of rebuilding the solution to determine the most useful features, leveraging click-to-action, migrating established applications, and integrating other products.
**Five key steps to portal success**

Before you continue reading, review the five key points for a successful WebSphere Portal project:

- Pay special attention to the process of creating reasonable requirements.
- Select the proper team composition.
- Choose the proper components and architecture for your infrastructure.
- Avoid unnecessary complexity.
- Create an effective testing plan and environment.

*Note:* The portal will become the most visible element in your infrastructure. The core support team will be held responsible for outages anywhere in the system.

### 2.2 Roles and responsibilities

There are many roles and responsibilities needed to staff a WebSphere Portal project. The diversity of knowledge required is so large because the portals are made to combine various systems and technologies. Therefore, your portal system will sooner or later touch all of them.

We usually categorize a portal project as an infrastructure project instead of a development project. Often, the only parts that need to be created are the code that holds the already available functionality together.

*Important:* Most pilot portal projects are overstaffed with developers. Portlet programming does not consume the bulk of the project.

We see the following roles most often in projects. Depending on the type of portal you are targeting, this might not be a complete list.

- **Project management team**

  There are presentations that suggest that you require the best of all best project managers to run your portal project. Indeed, many projects suffer due to insufficient project management. However, the failure of a project also stem from the some jobs being underestimated. The management of portal projects tend to be more complex and time-consuming so are often underestimated.

  Again, understand that your portal system touches many systems and departments of your organization. All of these systems and people need to be managed.

  *Note:* Consider appointing a team member that solely cares about “political issues.” If you have already done a portal project, you might already see the logic of this suggestion.

  If the project management team is not successfully able to manage itself and the portal project, your project is condemned to fail. Technical problems do not kill a project. At the most they protract it, but they can always be fixed. Portal projects can fail due to unsuccessful project management.
Architecture board

The portal system architectural work has challenges. We have seen very good success on projects that implement an architectural board consisting of architects of various disciplines. Usually, it does not make sense to wait for the great portal architect, because it is a rare person that can deeply analyze a network stream, a database layout, and a business process. We list a couple of architectural disciplines that we see in such a board. Except for the lead architect, they might not be required to be working full time at the project. Nonetheless, they need to be dedicated to the project during the time that they are required.

- **Lead architect**
  The lead architect makes the final decisions regarding architectural and technical topics. She needs to work closely with the project management team and is their most important person. Additionally, she (with the project manager) has to recognize and fill in the gaps where nobody seems to be responsible. The lead architect also has the job of working with the other architects for a good technical solution. In summary, she is the glue to hold the project together and needs to be treated like that. It is also important for the lead architect to delegate and work as a team member.

- **WebSphere Portal product architect/specialist**
  This person should be involved in the early planning of the portal about how to design the topology and similar things. Later, the development architect might want to kick off the project with the product architect/specialist, as well focusing on test and deployment scenarios. You might want to involve a person who has experience with the product and who you can involve in all product-related decisions. Depending on the size of your project, the product architect/specialist will not need to be on site 100% of the time. Most of the time, remote support might even be sufficient.

- **Network architect**
  This person is involved in the topology design, the build up and connection of the various environments, and finally the deployment of the systems. The network architect needs to be aware of all the issues that might arise from connecting the involved systems.

- **Database architect**
  The database architect is a role that depends on the needs of your project. Designing reasonable database schemas can be the best task to help performance.

- **User registration architect**
  We usually design or explain the Lightweight Directory Access Protocol (LDAP) schemas in place. This includes the LDAP search strings that can decrease the performance of your system. If you do not use a LDAP and you want to use an established user registration system that does not comply to current standards, this role is even more important. For example, we have seen systems that suffered because of the cascade of systems involved during the login process. The login request is the most important request on a portal system, so it must not be loaded with unnecessary complexity. Because your user data is extremely valuable, you might want to ensure that this is architected in a professional way by an experienced person.
– Security architect

Modern companies have many security guidelines. Setting up a WebSphere Portal project leads to new systems and new software that require security compliance. We have seen projects that were ready to go live but were put on hold, because it was unclear whether all security guidelines were followed correctly. We recommend having a person on the team at least part-time to ensure that your guidelines are correctly followed.

– Development architect/lead developer

This person makes decisions on a development level. This person needs to be in close contact with the lead architect and the project management team. This person leads the development team and recognizes early problems, for example, skill gaps in the team.

– Test architect/lead tester

This role can be two-fold. You need somebody who is in charge of planning and designing the load and stress test. Fill this role early in the process (see also “Non-functional tests” on page 83). Additionally, you need to have someone who designs the test scenarios to ensure that the portal provides the functionality as requested. This person needs to establish and administer the bug tracking tools to be used by the developers. A functional and a non-functional test architect is required.

➤ Developers

As stated in the beginning of this chapter, portal projects tend to be overstuffed with developers. Writing portlets is, however, not overly complex. Portlets are always based on the model-view-controller (MVC) principle. Portlets should be as simple as possible.

Portlet developers or a more specialized group need to be assigned to create the portal elements such as themes and skins out of the design guidelines.

Experienced developers are required to create and debug the connectors to your business logic or the business logic itself. It requires experience, because you need a good understanding of what can go wrong and how long it takes to get it right.

You will find some features especially for portlet development within the IBM Rational Application Developer tools. We generally recommend these tools (make sure that your developers have the right size development workstation, with a minimum of 2 GB RAM).

➤ Code maintainer

Within your development team you will require somebody to be assigned for code maintenance. This person not only manages the code repository (such as Concurrent Versions System (CVS) or IBM Rational ClearCase®), but also writes scripts for daily or weekly builds and deployment.

➤ Designers

In most projects, this is often outsourced to some external design company that has bright ideas about how to combine already available corporate identity and corporate design guidelines with a user friendly portal navigation and layout. Often, the output will be pictures and PDF documents, which the developers will need to transform into style sheets and JSPs. Make sure that the designers are available to the developers for requests that go beyond the graphics.

It is worth the investment to do intensive usability tests. Assume that your designers created a great looking portal design, but the fonts are so small people are unable to read it. This is the value of conducting a usability test to help identify those items to contribute to your design efforts.
Administrators

Because portal projects are really infrastructure projects where the administrators often play a key role, make sure that your team does not have a portal administrator who does all the portal configuration task, for example, creating pages and assigning skins to portlets by using the administration interface. You also need administrators who have access to your back-end systems. You might need additional rights for your database back end or for some user on one of your already established systems.

The portal administrator has to work closely with the code maintainer to get scripts ready for automatic deployments. Good administrators try to replace themselves with scripts.

Testers

For testers, you might again distinguish between functional and non-functional testers. As we describe in Chapter 4, “Deploying, testing, and maintaining a portal” on page 79, do not start testing too late in a project.

Support/maintenance staff

Think about the time after the project was launched successfully into production. You needed to have some people in place that performed maintenance until the next development iteration started. There might be a minor problem in a function somewhere that needs to be fixed. Additionally, you need somebody who is able to support the help desk for technical questions regarding the portal system.

This is often a task area that is outsourced. The support or maintenance staff gets the “leftover” documentation of the project, which is a bad start and might lead to difficulties.

Keep in mind that the response time and the quality of support requested from your portal users are significant factors in user satisfaction.

Ideally, some of the developers in the project team will become a part of the support and maintenance team.

Release manager

The release manager is normally a part of the project management team. The release manager performs the coordination of the dates regarding other things going on in your organization. For example, say that you want to perform a stress test over night, but that night half of the back-end databases were shut down for maintenance. It is not a good idea to perform this on the same day that you go live with your portal system, or when there is a major upgrade taking place at one of your systems somewhere in the back-end architecture. Just to experience these things the day you present your pilot to your CIO might cast a bad light on your project.

Although these can be obvious issues, they can happen over and over again; therefore, we recommend dedicating a person for this role.

Executive sponsor

A good portal project needs a person in place who has the power to make bold decisions, an important person to help clear things up if the project management team is not able to get things organized in a way that is most effective for the project.

Portal projects need organizations to communicate and collaborate in different ways to which they are accustomed. The executive sponsor, therefore, needs to facilitate and intervene as needed.
Building your team
Do not over-staff your team at the start of your project. Remember that *quality* over quantity is a key point.

The following summary emphasizes important items to consider:

- Designate an architecture team and assign them broad authority over the infrastructure.
- Insist on senior-level team members as the core architects.
- Building Web services and incorporating connectivity to complex back-end systems is well-suited for an experienced professional.
- Delegate documentation, theme building, and basic portlet coding to junior-level members.
- Communication skills are paramount to the success of your project. Enlist an excellent project manager.
- Include an experienced WebSphere Portal architect on your team; consider engaging IBM Services for your first project. Refer to 2.3.3, “Solution design” on page 21.

Understand that the team configuration will change over time. Post-pilot, the focus will shift from infrastructure to development. This is the time to augment the development team and optimize the architecture team.

2.3 Requirements process

The requirements process that you have in place might not entirely fit the needs of a WebSphere Portal project. Always remember the special configurations of a portal project and most importantly try to add the least complexity for the first release. Add less important features later.

2.3.1 Define the goals

A business case that supports the funding of the portal project is extremely important. After you have your sponsor, there will be the need to *sell* the concept within the organization, whether to the board of directors, a technology council, or other lines of business. These decision makers will want to understand the cost that is required to make the portal ready for reuse. They will want to understand the anticipated revenue that a portal will generate and a break even rate analysis.

One asset that IBM provides is the Business Value Assessment for Portal engagement. This can help provide the high-level direction you need to create a vision, identify success criteria, and build a preliminary cost and benefit analysis. Refer to the following link for more details:

http://www.ibm.com/industries/travel/doc/content/bin/050622_Business_Value_Assessment_for_Workplaces_brochure.pdf

Request a Solution Assurance Review

A Solution Assurance Review can improve your success rate. It will help you to minimize problems with the installation and configuration and increase the odds of completing your project on time.
A Solution Assurance Review is a mechanism for validating the technical design of a proposed solution. It takes the form of a technical inspection of a proposed solution by parties other than the solution designers and carried out before it is implemented.

A Solution Assurance Review is a technical inspection of a completed solution design. IBM technical subject matter experts who were not involved in the solution design participate to determine if the solution will work, is the implementation sound, and will it meet the end user’s requirements and expectations. The outcome of this exercise is a report that describes the known risks of the project and provides a list of ways about how to mitigate those risks. This can include identifying a fix pack, suggesting certain configuration settings, or providing a list of education classes.

Contact your local technical sales team to request a Solution Assurance Review. Refer to Appendix D, “Solution assurance checklist” on page 113 for a complete checklist. For this process, complete the checklist and your local technical sales specialist will submit it for review on your behalf.

**Leverage the portlet sourcing method**

If you have Web sites or portals that you intend to combine, think about a process similar to portlet sourcing. We describe this in Appendix C, “Portlet sourcing” on page 105.

### 2.3.2 Project plan

Because the portal will be the most visible element in your IT infrastructure, you need to ensure the success of this component over all others. If the portal fails, your infrastructure will be perceived as a failure.

See Appendix B, “Sample portal tracking worksheet” on page 103 for a sample project tracking plan from the IBM Services team. You can use this project tracking plan as a starting point. This document contains tasks, people, and time lines based on the experience of hundreds of projects performed by dozens of IBM consultants. Although it does not contain every task that you might need in your project, it is a very good starting point for your project manager. Use this as a baseline; then, refine tasks and time lines as needed.

**Important:** Set the right expectation. WebSphere Portal is not a simple plug-and-play application. It is a horizontal portal framework. Treat it like a complex infrastructure project. Again, plan four months at a minimum for the first pilot release.

### 2.3.3 Solution design

At this point in your project planning, you need to document standards for the project. These include, and are not limited to, a change management process, documentation guidelines, and a testing process. It will also be your first attempt to establish a road map for the project including implementation phases.

Consider starting this phase of your project with a 3-to-5 day Architecture and Design Workshop from IBM Lab Services.

The main objectives of this engagement are:

- Review all major aspects of the portal project.
- Identify gap items that introduce project risk.
- Get recommendations for mitigating gap items.
Here is what one of our customers had to say about the architecture workshop:

“After two failed attempts to deploy WebSphere Portal ourselves, this three-day workshop helped us to resolve the issues that we’ve been struggling with for three years.”

An Architecture and Design Review can help save you from project failures. Make sure that all stakeholders attend portions of this effort. Repeat this step whenever you have a major new initiative or new teams involved. For more information about Services offerings, refer to the following document:


2.4 Determine and reduce the complexity

The complexity of projects is often far too high. All encompassing projects often fail. Start with a base portal installation and choose a few components to deploy at first. For example, configure the portal for your database and LDAP servers. Then, add connectivity to a back-end application.

Build your portal from the base components first. Only install added functionality when the infrastructure is solid. Use an iterative process and load test along the way. This will simplify the process of problem isolation.

Do not attempt to create a reference architecture on your first project. Let your reference architecture grow in a planned fashion.

**Tip:** Pick the right tool for the job and drive business value with simple technologies.

To ease complexity in a proof-of-concept, you might also use the Web Clipping and iFrame portlets in moderation. They are quickly established, but not the best choice as long-term solutions. Therefore, use them, if at all, only in moderation.

For a proof-of-concept, it might also be reasonable to leverage the Web Clipping features. However, be careful here, because you might be able to write a new portlet in the time it takes to work out certain challenges of Web Clipping.

Generally, we recommend that you do not use Web Clipping or iFrames in a production scenario. Just assume that the Web sites you are reusing will change without notification. For more information regarding this topic and a pointer to a dedicated white paper, see 3.2.3, “Markup generation” on page 63.

There are meanwhile a couple of tools and add-on products available that can help you to get things done pretty quickly without worrying about the complexity involved.

You might, for example, use WebSphere Portal Application Integrator to create portlets that connect to a back-end database. For some projects, it is a good idea to use rapid application development tools, such as Bowstreet Portlet Factory or AlphaBlox.

It is not possible to give a general recommendation of what to use for which type of portal because of the diversity of tasks and challenges involved. We can only say that it is worthwhile to spend some time considering them. After that, you have to trust your technical experts on the team to choose the right technology for their job.
However, there is one exception: if your developers or architects suggest building a new framework for the portal. WebSphere Portal should have most of the front-end frameworks that you will need. Adding more will not make things easier; it will only delay your solution.

We often get requests to describe how and what to reuse of existing Java 2 Platform, Enterprise Edition (J2EE)-based Web sites for a portal solution. The most reasonable thing to reuse is the knowledge of your team members. This does not just include your developers, but also architects, project managers, administrators, and so on. Your team is a big asset.

From a pure Java code point of view, it is often better to throw away current servlet-based code and frameworks instead of trying to reuse it. If you have a model-view-controller (MVC)-based, well-layered application, keep the business logic. Throw away the controller and user interface (UI) logic. Obviously, you might want to keep cascading style sheets (CSSs) and liberally cut and paste some JSP code.

If you then create new portlets, remember that the best portlets are the most simple ones. Avoid rebuilding something that is already available as a portlet service or within frameworks, such as Struts and JavaServer Faces (JSF).

Tip: Spend more time in the WebSphere Portal Information Center, the Apache Jakarta pages, or on Google, instead of coding already available functionality on your own.

Do not fall into the super portlet trap. Because portlets are so simple to implement, multiple portlets are not as onerous. Simple and multiple portlets are better than super single portlets. The use of portlets forces you to reduce the complexity of user interface and controller logic and pushes complex business logic to where it belongs.

2.5 Defining non-functional requirements as part of service level agreements

In addition to the functional requirements of a portal project, their counterpart, the non-functional requirements, are just as important. These types of requirements are necessary as part of the foundation in setting forth the goals of your project. The non-functional requirements are usually defined in service level agreements (SLAs). A service level agreement defines the agreed to service levels or measurements (availability and performance objectives) by which the solution will be supported in the organization. For more information, see:

http://en.wikipedia.org/wiki/Service_Level_Agreement

Therefore, the non-functional requirements are just a part of the iterative process to create proper SLAs. They can change over time. For example, the business need of the portal changes, and as a result, the response time needs to be shorter. In this section, we concentrate on portal-specific criteria that are very important for a solid portal implementation. We do not cover SLA topics, such as the response time of the support staff during weekends in the event of an outage, here. Nevertheless, these standard IT project items need to be in place and well covered.

2.5.1 High availability

There have been quite a few projects where we found this number randomly picked instead of clearly calculated, or the numbers had been labeled on a similar project and were, therefore, meant to be reused. These methods not only are a waste of time and money, they can also lead to disagreements between the involved parties.
Availability is technically defined as a result of the mean time between failure (MTBF) and the mean time to repair (MTTR). Figure 2-2 shows the mathematical equation.

\[
\text{AVAILABILITY} = \frac{\text{MTBF}}{\text{MTBF} + \text{MTTR}}
\]

Figure 2-2  Equation

A failure or repair situation is not the only factor that influences the availability factor. Assume that your portal environment is not available on the network every night for one hour to perform a backup. This adds up to 365 hours or 15 days and 5 hours a year, which leads to a maximum availability of 98.83% (23h/23h+1h).

It will be up to you to explicitly exclude scheduled outages from your availability factor, and if so, remember to include a discussion on the handling of scheduled outages within the organization. We often see that the IT department receives a certain number of hours per year to perform maintenance, usually within a certain time frame at night. Beyond this, it is reasonable to designate a person or define a process where you can apply for an exception to the scheduled outages, for example, to apply a security patch to the system.

Because a portal is by definition the front end of a number of systems, its availability is influenced by all of the supporting back-end systems. The overall availability of an application is the result of the availability of all components multiplied. A theoretical example is:

- Availability of the user's Web browser (98%)
- Availability of the user's DSL connection (98%)
- Availability of the Internet backbones (99.99%)
- Availability of the Web site's firewall (99.5%)
- Availability of the Web server (99.8%)
- Availability of the Portal server (99.6%)
- Availability of the network in the data center (99.99%)
- Availability of the operating system (99.99%)
- Availability of the database (99.5%)
- Basic data center availability, covering disasters, and so on (99.999%)
- Perhaps more

This leads to an overall availability of less than 94.5%.

From a user's perspective, all of these components are required to allow the user to leverage the portal application; it does not matter to the user why the portal is not available.

In your discussions, make clear that you are defining the availability of the portal system itself, and explicitly exclude the other components involved. Often, this argument will not work because the owner of the portal might also not care about which component fails. You should at least agree to a point in the system where it can be measured, for example, available from a certain point in the internal network. Try to take responsibility for the portal piece in the system, and then negotiate proper SLAs with the other system owners.

Needless to say, costs will determine the high availability factor to which you can commit. Redundancy of hardware and software is not a trivial expense.

The problem of measurement is an obvious one regarding response time. However, it does apply to all the topics that you define in your service level agreements. Whether they are
technical or non-technical, you need to know in advance and describe in your SLAs how you measure the numbers.

The best advice we can give you is to clearly define the term *availability* in your project plan to mitigate any risks and allow you to reach your goal. We are fully convinced that no matter what high availability number you target, WebSphere Portal is now mature enough to comply.

### 2.5.2 More about non-functional requirements

As part of the IT organization supporting a portal, you need to negotiate wisely with business users about the items we discuss in this section. It is important to set expectations up front. Remember, the accuracy of the information needed to adequately size a portal might be difficult to provide in the planning stages of your project. Exact numbers for your environment can only come from actual usage in a real-time setting. Explain to the business users that this is an iterative process.

**Parameters to define the non-functional requirements**

Here, we mention the most notable parameters to be considered. This is not a definitive list. Other factors such as session persistence, Secure Sockets Layer (SSL), and maximum processor utilization also have an impact.

In addition, note that we discuss the parameters more to provide you with a reasonable Techline sizing instead of negotiating reasonable service level agreements.

**Number of registered users**

This number is usually very easy to provide; however, it is often a useless number. There are certain projects where this number does matter. For example, if you have your users in an LDAP database, this number factors into the execution time of LDAP search paths. A good example of this is the use of the People Finder portlet.

It also plays a role in the estimation of hard disk capacity of the database and might even remotely affect the general non-functional requirements of the database. Generally, the effects of this number are overestimated.

**Number of concurrent users**

The number of concurrent users describes the number of users that are currently active on the Portal server. This is misleading, because it is not the same as *activity*, and it must not be assumed that you can calculate out of it the amount of work done on the portal. Therefore, the number on its own is not very helpful when calculating the number of CPUs required for your portal system.

The number of concurrent users is important when calculating the Java heap size. Set the Java heap size to facilitate efficient application execution, including garbage collection. Garbage collection begins to become excessive when the Java Virtual Machine (JVM™) cannot find sufficient memory to execute a request. Therefore, garbage collection can be one of the biggest bottlenecks for an application. Depending on the size of your portal system, a close assumption of the required Java heap size is also important in order to calculate the number of virtual machines (VMs). We discuss how heap size might become an issue in “Where to cache?” on page 70.

To calculate the required heap size from the number of concurrent users, we need to know the amount of memory a single session requires. Without knowing ahead of time what size a single session will be, we are again at a disadvantage. The size of a single user session in a clean portal out of the box is about 5 KB. How much a single session in your portal system...
will require depends your applications. See 3.4.2, “Sessions” on page 76 for more information regarding session management in your portlet applications.

The number of concurrent users can be drastically altered by simply altering the session timeout. A two-hour timeout might show six times the number of concurrent users in comparison to a 20-minute timeout.

**Tip:** Session timeout is closely related to the calculation of concurrent users.

**Think time**

Think time is the amount of time a logged-in user pauses before requesting a new page view. Note that in a Techline sizing exercise, the default is 30 seconds. Sometimes, it is impossible to get an accurate number any other way but empirically. More than likely, you will need to have your portal application up and running in order to get a good estimate for this value. Remember, this is an iterative process.

If providing a best educated guess is not possible, you can enable a pilot with a selected group of users. Selected user-group pilots do occur quite frequently during big portal projects. These pilots, in conjunction with usability pilots, offer solid feedback to the project team. Consider think time as an important component of your stress tests.

As a general rule, we see think times higher in transactional portal sites than in content-based portal sites. Think times in intranet portals appear to be at the low end, low-cost e-commerce sites fall in the middle, and banking sites are at the high end. Another approach is to duplicate the think time from similar Web site as a starting point.

Without having the average think time, it is almost impossible to get an accurate picture of the requests per second variable.

**Requests per second**

A request is anything that causes a Web page or information to be regenerated. This number is requested for a Techline sizing. You usually need to calculate the requests per second from other provided information. This is a very important number to have, because it helps to calculate the CPU load that you will experience.

Be careful if this number is just given to you. It might be purely a best guess and not be based on a solid calculation. Remember that it is the duty of a good IT architect to detect if input from the business side is inaccurate.

**Logons per second**

This number is valuable in calculating the CPU load. However, it is more important for accurate CPU sizing of the LDAP and database servers than for Portal server sizing. In WebSphere Portal V5, there has been much improvement in performance for the logon task. It is still quite an intensive operation for the LDAP and Portal database servers.

**Percentage of anonymous requests**

An anonymous request is a request coming from a user that is not authenticated and thus has no session within the portal environment. The big advantage that you can gain out of an anonymous requests is that these requests enable you to perform massive caching.

**Tip:** Try to avert the requirement to turn on public sessions.

Having no session, they also require less memory in the heap. Generally speaking, anonymous portlets provide less functionality and are usually simpler, but also consume less
resources in contrast to a portlet that performs a transactional requests to a back-end system. Anonymous users do not log on and thus do not generate a logon requests, thereby easing the load on both the LDAP and portal databases.

**Page response time**
This is a number perceived by the end user as the time it takes the Portal server to respond to a request. For more about response time, refer to:

http://en.wikipedia.org/wiki/Response_time

The page response time is a very important number to negotiate. More times than not, the number supplied here is unrealistic.

Assume for a moment that your application cannot be improved any more. You can decrease the load on a CPU by increasing the response time during page execution. A portal CPU can serve up to 10 times the number of requests for a 40 second response time than it can for a .5 second response time. Both numbers are too extreme, but provide you with an idea of how these two relate to each other.

**Important:** The more load on your portal, the longer the response time.

To allow for a more reasonable sizing, you might want to plan for a generous response time during peak hours. This creates a more robust response time in an average load. Accurate, realistic numbers are very helpful in sizing for the portal system.

**Techline sizing**
Sizing the hardware needed for your portal implementation is accomplished through the Techline Sizing Center.

A sizing estimate is an approximation of the hardware resources required to support a Web solution implementation. It is a pre-sales effort based on information available at a point in time, providing an entry into understanding your hardware requirements. Customers’ actual experiences will vary from the sizing estimate for many reasons, including network activity, application design, and availability factors. The degree of variability can range from small to very significant. It is important to understand that the sizing estimate is a pre-sales effort mainly based on benchmark performance data; it should not replace capacity planning for installed systems. You can use the sizing estimate for preinstallation planning; however, when you are in the process of implementing your solution, work with an IBM Capacity Planning consultant to monitor and predict the ongoing resource requirements. Customer results can vary, and IBM assumes no liability for actual results that differ from the sizing.

The accuracy of this sizing estimate depends on many variables such as the accuracy of the information provided by you, the customer, (we recognize that it might be difficult to provide exact numbers on your environment) and the actual usage of the system in a real-time environment. View sizing as an iterative process.

If you have accurate numbers for the sizings mentioned earlier, the Techline sizings are usually quite accurate. Nevertheless, figures might be misinterpreted, so never trust just one number, always double-check for verification. You can request a technical review from a software group or one of the closest IBM Lab Services portal architects. A remote review might help to point to important overlooked planning items.

Some blame the Techline sizings if the portal system does not perform. Unfortunately, this is not the case. The Techline sizings are based on an algorithm that has as a basis data from hardware vendors and the software labs. Therefore, these are real simulation numbers, and you have to try to remove the ingrained bias to inflate. In particular, remember that feeding in
wrong, misleading, or inaccurate data naturally leads to wrong, misleading, or inaccurate output.

To request a Techline sizing, call Techline Phone Support at 1-888-426-5525.

2.5.3 Calculating the costs

We begin this section with a quote from Albert Einstein, “Everything that can be counted does not necessarily count; everything that counts cannot necessarily be counted.” This is how we approach calculating the costs of owning WebSphere Portal. IBM WebSphere Portal drives sales, employee productivity, and customer satisfaction while helping to reduce employee turnover, travel time, and training expenses. WebSphere Portal simultaneously reduces costs and improves service. The challenge is not only finding return on investment (ROI), but helping you choose the best ROI model.

Return on investment

At some level, all business decisions will have to look at cost. For example, there is a cost to implementing a search engine and there is a cost associated with people not finding the content that they are looking for on your site. When looking at adding a search engine to your environment, consider the cost of adding the solution from two perspectives: the cost and the time to implement.

From the cost perspective, the purchase price of the software might be significant. Most products require some form of maintenance for ongoing support and upgrades. Make sure that you understand all of the costs associated with the initial purchase and ongoing maintenance of a given solution. From the time to implement perspective, after you purchase a search product, you will need to install, configure, and implement the solution to work in your environment. This cost is often difficult to define, but underestimating this effort can lead you to flawed ROI numbers.

When you bring these (and other) factors into the calculation, it is possible that the return on your investment is not high enough to invest in a search product. If this is your outcome for a portal decision, consider performing a return on investment study. Refer to the following link a tool to evaluate cost factors and revenue benefits:


We use the term “total ROI” to denote the combination of the benefits (the increased ROI due to increasing benefits, keeping costs the same) together with the ROI impact of cost avoidance.

Real cost reductions

How do you quantify hard dollar cost savings when considering a WebSphere Portal implementation? At IBM, we migrated our intranet site to WebSphere Portal in April 2004. As part of that implementation, we made collaboration tools available to a workforce of more than 320,000 users, 40% of which do not work in a traditional office setting. IBM employees host more than 10,700 e-meetings per month and log more than 141,000 person hours in e-meetings. IBM conservatively estimates that based on the number of attendees and the number of Web conferences, we save $10 per Web conference due to avoided travel costs. This figure takes into account that not all e-meetings required travel. We average 10,700 Web conferences per month with 6.8 attendees per Web conference.

Important: IBM estimates that it saves $7,898,000 in annual travel costs.
IBM reduces survey administration and analysis costs through the use of Web forums and online survey tools, also a part of our portal, and generates an increased response rate. We saw a reduction of five surveys per year at an average cost of $2 per survey.

**Important:** IBM estimates that it saves $3,192,730 in annual survey administration costs.

Determine if you can apply some of these metrics for your business practices, and along the way move, your portal mission from an outward bound communication tool to a platform where real work gets done.

### 2.6 Initial technical considerations

This section provides a good overview of the technical considerations to consider while planning a WebSphere Portal solution. We often see projects where this has been done too late or too early.

Too early is, for example, if there is no business justification for a portal yet. You would have no reasonable basis to do your planning. We see organizations declare WebSphere as their middleware and purchase WebSphere Portal for their emerging technology or research and development staff to experiment. The missing link here is a line of business with a need for a portal application.

On the other side, it is too late if the project plan is already established and the time line fixed. You would have no room for your technical considerations to still influence the project. Your only course of action is to make the best out of it, which is a bad start for an infrastructure project.

#### 2.6.1 Topology planning

There are many presentations out there describing various topologies. This section provides some typical samples, mentioning their advantages and disadvantages.

**Standard configuration**

The configuration as shown in Figure 2-3 on page 30 is the general best practice to build up a WebSphere Portal environment. This describes a classical 3-tier portal layer topology, including a demilitarized zone (DMZ). With the exception of the single server developer installation, this is probably the configuration that is most often tested in the software lab.
The first tier is behind the outer firewall, sometimes also called domain firewall because it secures your domain. This should be a fairly naked environment, applying security using the concept of least complexity. You would never deploy the Portal server in the DMZ. In addition, do not deploy any other JVM-based systems there. Keep your DMZ as clean as possible.

For security reasons, we generally recommend some hardened UNIX- or Linux®-based system.

You can usually use some low-cost servers for this layer, but do not use a cluster on this layer. The machines are inexpensive and software, such as the IBM HTTP Server, do not require license costs per CPU. While this layer needs to be load balanced by some Network Dispatcher, the load balancing of the Portal servers is performed by the WebSphere Application Server plug-in that runs within the HTTP server.

We do not include the Network Dispatchers in the illustration, because they do not work on the upper layers of the network protocol (for Network Dispatchers, it is sub-IP level, which equals OSI-Level 2), and therefore, a network administrator is usually in charge of selecting and configuring them.

It is reasonable to include a DMZ. The consequences of not having a DMZ can be more than the cost of a DMZ.

**Important:** The DMZ is also a must have concept if you are planning an intranet portal. More security threats happen from inside a company than from outside. All connections from inside or outside have to be terminated in the DMZ.

The backside of the DMZ builds the DMZ firewall, sometimes also called a protocol firewall. Here, do not have the same ports open as you have on the outer firewall, where you should allow only access on the ports 80 and 443.

Figure 2-3 shows a standard two server Portal server cluster. See 2.6.2, “To cluster or not to cluster” on page 35, where we discuss if and how you should architecture your portal cluster environment.
The back end shows various databases and other systems. Those systems can, for example, include host systems, custom-developed systems, or even systems with proprietary protocols. It depends, therefore, on your already available topology and on the type and the goals of your portal application. We highly recommend using a LDAP system for the user registry. Depending on the amount of data you expect in your LDAP, make sure that the implementation you select is able to cope with your requirements. If you want to use a LDAP system that is already in production, make sure that it can handle the additional load that comes through the portal and also make sure that you have a mirror system of the target LDAP available for your stress tests. See also 4.1, “Test” on page 80.

It is not unusual to also have the back-end database or the other systems, or both, separated by one or more firewalls. Discuss this with your network and database administrators. In general, think security first.

**Alternative to the standard configuration**

There is an alternative to the previously described standard configuration that we do see more and more often in the field.

In Figure 2-4, the Web server is replaced by a caching proxy. The use of a reverse proxy allows much more cached data than, for example, the Web server plug-in can do on its own. Similar to a Web server, reverse proxies also allow to host static public content. In addition to the technical features that come with a caching proxy, such as WebSphere Edge Server (http://www.ibm.com/software/webservers/edgeserver/index.html), there are sometimes pure organizational reasons. Network and data center administrators sometimes do not any software components designed by application architects running in their DMZ. For the most part, this is because of the fear that it might get harmed from a security perspective. Therefore, you might have to move your HTTP server with the WebSphere Application Server plug-in out of the DMZ.

![Diagram](image.png)

*Figure 2-4  Alternative version of the standard configuration with a caching proxy*

We do not recommend that you leave out the Web server at all and rely on the load balancing that you will get out of the DMZ. The additional hop of a Web server should not influence your performance negatively, and the computing resources required should be covered. It is worth the resources to have the proper balancing ensured in case a WebSphere Application Server goes down for maintenance or due to a program failure.
We did not find any performance comparisons of accessing WebSphere Application Server directly versus having a Web server still in place. We did not, however, discover many customers who use this configuration, and again, we do not recommend it.

There had been internal tests done with a collocation scenario to see if you can gain performance by increasing the priority (priority boosting policies and priority levels) of the JVM. Having all components in place optimally tuned and for a given fixed client workload, it did show a marginal performance improvement. Trying the same configuration on a customer’s portal showed no effect or a negative effect. It is likely that the reason was the sole CPU power to handle the connections, not the throughput.

Although many articles and presentations might give you a different impression, we do not believe it is a problem to have IBM HTTP Server and the Portal server on the same machine. Articles that recommend against this might be simply reminding you that you must not forget to build up a DMZ and you must not deploy your application server in the DMZ. We can fully comply to this configuration, which is a best practice.

From a performance perspective, there is no reason not to run a lightweight Web server, such as IBM HTTP Server or the open source Apache, on the same physical machine.

Note that this is a very general statement and that there might be reasons, such as a high load on static content from the Web server, that might lead to a different configuration. The following section describes an improved standard configuration.

**Improved standard configuration**

In the two previous sections, we describe two configurations, each with a drawback: either no caching proxy or the collocation of the Web server and WebSphere Application Server. If you need to choose between them, take a close look at the work they are supposed to do in your environment. Can you leverage at all the advantages that a caching proxy gives you? Are you able to push all work away from the Web server so that it does nothing else but host the WebSphere Application Server plug-in? Sometimes, you might not have the chance to get your preferred way due to restrictions from your host.

You can improve your topology with a separate Web server layer, as shown in Figure 2-5 on page 33. The advantages include:

- Easier maintenance (because there is then only one product per node)
- Greater flexibility in your architecture for later changes, because both components in are in one place and one is fully accessible because it is not in the DMZ
From a pure technical point of view, it is hard to understand why leaving out these clustered HTTP servers is a worthwhile topic to discuss. Because hardware is so cheap and license costs are not an issue in this layer, adding a couple of more nodes should not be an issue. But the experience in the field, which shows extremely tight budgets, does prove that it is sometimes an issue. Just be careful not to run into the trap of saving $2,000 for hardware or hosting and spending an additional $20,000 for consulting services.

**Example with calculated single points of failure**

The previous examples all target a standard configuration for building an environment without any single points of failure. You do not want to have a single point in your system that can cause a failure of the overall system. Therefore, having no single points of failure in your system is a core element for any reliable system design. Because a reliable system is a logical and natural requirement for a portal system, the best practice for a portal is to avoid any single points of failure in your system. For more information about single points of failure, refer to:

http://en.wikipedia.org/wiki/Reliable_system_design

However, because portal systems are not built because of architectural designs but because of rational and clear business needs and decisions, there are cases were the general best practices do not apply.

Naturally, you want to provide a reliable system. However, it all depends on the service level agreements listed in the contract. It is up to the parties involved to define clearly what they understand by the term **reliable system**.

Let us assume that the diagram in Figure 2-6 on page 34 describes a public information system, such as the portal of a city, containing information about events and other things going on in the city. Assume further that all access to this portal site is anonymous, with required logins or sessions. We can agree that we do not require a WebSphere cluster with session failover here, because the WebSphere administrator is the only one able to take advantage of session failover. The same applies to your databases and LDAP server. Only your content management system (CMS) should be able to reliably deliver its content.
If we add a free-of-charge service to the scenario, for example, to give the users the ability to write and publish their own blogs, would that lead to a different situation? If your LDAP has a failure, users will not be able to log on, and if the database or the portal node on which the users currently resides has a problem, the users will lose all their work. Therefore, consider how much you are willing to invest for the reputation of a free-of-charge service.

Going even a step further, assume that you can log on to this portal as a representative of a company to add commercials such as banners. Here, we always want the users to be able to submit their commercials. In this scenario, we are concerned about how high the probability of such a problem is and how much business do we lose during such a situation.

For more information about this topic, see 2.6.2, “To cluster or not to cluster” on page 35.

**Example of a high-end configuration**

In contrast to the previous examples, the diagram in Figure 2-7 on page 35 describes a rather complex scenario. This is not an unusual scenario for bigger customers, and indeed, more complex configurations exist. However, we do not cover this configuration in this best practices paper.

If you plan to have a similar environment, we recommend that you have an experienced WebSphere Portal architect on board. WebSphere architects are not, however, the target audience of the Redpaper. They should already know everything stated here.

Even more complex configurations, such as clusters that span multiple geographical locations or the federation of multiple portals, are also out of the scope of this paper.
Chapter 2. Planning a portal

2.6.2 To cluster or not to cluster

Whether or not to cluster is a common question, especially in the small-to-medium business market. Due to the typically higher requirements on availability, there is often no way around clustering for bigger portal solutions. Additionally, if you read through the articles and presentations about WebSphere Portal topologies, you might assume that it is just natural to have a cluster. And indeed, most customers have clustered systems. With respect to WebSphere Portal, the process of clustering your servers has been greatly improved although a portal cluster is more difficult and time intensive to implement and to maintain than a non-clustered solution.
There are two main reasons to create a cluster (and they compliment one another):

- To scale horizontally, vertically, or both

  There are borders in growth as long as you deploy a single JVM. We do not recommend that you give a JVM more than 1.5 GB of heap. Although we have seen the JVM behave reasonably across four and more CPUs, as a general rule, we recommend having one JVM per two to four CPUs.

- To reach a higher availability

  If you just have a single JVM, there is nothing you can do to improve the availability of it (except to improve your application, and we assume already it is working well). If your calculated availability of a single server is not sufficient, consider the costs to increase it. If your budget it too tight for that, you might want to change the service level agreements accordingly.

  **Important:** If one of these reasons apply to you, think carefully. Do you *really* need a clustered environment?

If you need a cluster setup, keep in mind that a cluster requires special attention in all of these areas: planning, implementation, deployment, and ongoing maintenance. If there is a reasonable way to work around the setup of a clustered environment, do not be afraid to choose this approach.

Assume that the scope of your project does not require a clustered environment, but it is very likely that it might in near future. In that case, our experience has told us that technically there is an advantage to start clustering as early as possible. As a developer, you will be able to detect very early problems that only occur in clustered systems. For example, putting a non-serialized object into the session in a single server scenario is not a problem. If the session needs to be shared between various nodes, it presents a problem.

If it is not possible to start clustering now, from a best practices point of view, you can start with a single server in the first iteration of the portal. Do not worry too much about the problems that you might encounter in the future because there is usually enough to do in the beginning of a new project. If at a later time you need a clustered environment, there might be other changes going on as well. Therefore, at that time, the portlets might need to go through a regression test (see 4.1.3, “After going live” on page 88). If you use a lightweight development process, you should not require additional time due to such a step-by-step approach.

The most important thing to remember for project planning, whether you cluster now or cluster later, is that there is a cost associated with it. The costs are not limited to hardware and software licenses, but include project planning, staff development, and regression testing.

### 2.6.3 Horizontal versus vertical clustering

Because this is a paper about best practices, we also discuss the common question of horizontal versus vertical clustering in respect to WebSphere Portal.

Lately, we have seen less vertical clustering in the field. There are several reasons for this:

- Horizontal clusters might be required anyway.

  There are physical limitations to scaling your hardware regarding CPU and memory usage.
WebSphere Portal scales linearly.

Tests have shown that WebSphere Portal on its own scales linearly, which is a result of a considerable amount of work by the portal performance department. From project experience, we know, however, that this is not always true with the applications built on it. Garbage collection can slow down your system, depending on how often it occurs. To work around this, apply reasonable changes to the JVM parameters. Consider enlisting the help of IBM support, because this requires technical expertise.

Do not try to develop an application while already having a set of tuned JVM parameters fixed in your mind. You might find at deployment time that you cannot set them or with testing that they do not achieve the expected results. This might result in lost time and a rework of your application.

Virtualization technologies.

IBM @server pSeries® servers offer the ability to create multiple logical partitions (LPARs) on one powerful machine. From a portal perspective, each of them looks like a real physical machine. This allows much better management of the available server resources. For example, instead of running three vertically clustered nodes on an 8-CPU machine, you can now create the same number of LPARs and cluster it horizontally. This shifts more of the administration to managing the operating system and reduces the burden from WebSphere administration. Tasks such as the portal installation and upgrades are simplified. For example, port conflicts are eliminated. Growth management becomes easier.

In your first iteration, your pilot release, you might, for example, want to put on a single machine development, test, and production environment. As your system grows, you can dynamically add more LPARs.

Blade servers.

Blade servers typically come with four or less CPUs. Because WebSphere Portal scales linearly, consequently there is usually no need for a vertical cluster.

For more in-depth information about clustering, refer to the WebSphere Portal for Multiplatforms Information Center, available at:

http://publib.boulder.ibm.com/infocenter/wpdoc/v510/index.jsp

2.6.4 Security

Everybody agrees about the importance of security and that it is worth an investment to get it. But what is it? The following definitions from the free encyclopedia Wikipedia (http://en.wikipedia.org/wiki/Security) provide some interesting statements:

- “... as a technical term “security” means that something not only is secure but that it has been secured.”
- “When our expectations are met once and again, despite of errors, catastrophes and attacks which in principle could prevent our expectations to be met, we can say that security has been met.”
- “We can prove that there has been a security failure, but we can’t prove that there hasn’t.”

Security concepts

There is more about the topic of security than whether to use SSL or not. A valid, elaborated security concept is required for each portal system, no matter how big or small it is.

It is beyond the scope of this paper to discuss security concepts in detail. However, we describe a number of items that you need to control to ensure they are already included in
your security concept. Additionally, it might be helpful to consult IBM Redbooks and Redpapers that focus on security, available at:

http://www.redbooks.ibm.com/cgi-bin/searchsite.cgi?query=Security

Important considerations for your security concept include:

- Single sign-on scenarios and their implementation

  Because this is one of the most discussed items about security on portal systems, we discuss this in a separate section, “Single sign-on scenarios” on page 41.

- SSL encryption

  Send any personalized data encrypted through networks that are not secure such as the Internet or your even your intranet. This depends on the data that gets transmitted. An employee’s salary sheet is important to encrypt for transmission on your intranet. Some company presentations might be less important (depending on the information that they contain).

  You might wonder whether it is an advantage to have all data SSL encrypted. That might lead to a reduction of complexity. One of the main reasons not to do this is the performance decrease that goes with it. The encryption of data is a performance-intensive task, both on the client and the server side. This again might lead to a less satisfactory user experience. One of the main reasons not to use SSL encryption at all are costs. SSL certificates and their maintenance are expensive. SSL termination often requires additional hardware.

From situations with customers’ portals, we recommend following these best practices:

- Test to ensure that the logon credentials are encrypted while transmitted.

  In this portal, the anonymous pages were delivered unencrypted. The customer followed the best practices and made sure the login page was delivered as an SSL secured page. This gives users a sense of security while providing their credentials. As the browser showed in its location header, the authenticated pages after logon were also delivered SSL secured. The POST request itself, however, which sent the user credentials to the server, was put together with JavaScript™ and did not use SSL, but was sent through a clear HTTP request.

  No matter what your browser shows you, make sure that you test with a network sniffer, for example, Ethereal (http://www.ethereal.com), to be sure what really got transmitted over the network.

- Request your SSL certification early.

  A situation we detect quite frequently is that if there is not a dedicated security architect on the team, who would naturally think about, nobody seems to be responsible for SSL certification. Discuss this issue early if you require a certificate from a certificate authority (CA), such as VeriSign (http://www.verisign.com) or Thawte (http://www.thawte.com), and prepare for the proper requests early. If you are running an Internet portal, you always want to get a certificate from an official certificate authority. For your development and test systems, you might want to start off with a self-signed certificate.

  If you are building up an intranet portal, there might be already certificates in the company that you can reuse. You might also use your self-signed certificate, but for better user satisfaction, use a cross-signed certificate of an official CA.

  **Note:** Discuss this early. More people than you thought will be interested.
– Secure your SSL certificate.
    Make sure that only authorized people have access to the SSL certificate itself. Securing the certificate while still distributing it over the terminating servers is a challenging task that is frequently underestimated.

– Use SSL ID tracking.

SSL ID tracking is a convenient method supported by WebSphere Application Server for tracking a user session using the SSL ID instead of using a session cookie. Because the policies of some companies or the private policies of some users prevent the use of cookies, this is a good opportunity to solve this for certain portal systems.

A couple of years ago, this question regarding the usage of SSL ID tracking was always "no," with the answer that Microsoft Internet Explorer does not allow this technique. And indeed, on the Microsoft support site you can find a document (http://support.microsoft.com/kb/q265369/) that explains how Internet Explorer does a full SSL session renegotiation every two minutes. A periodic full SSL session renegotiation is contrary to the use of the SSL ID for session tracking. Because not supporting Internet Explorer was usually not an option, it did prevent the use of SSL ID tracking and made those customers considering it change their concepts. New tests show that at least starting with Microsoft Internet Explorer Version 6, Internet Explorer, and all new browsers (such as Mozilla, Firefox or Opera), appear to allow the usage of SSL ID tracking.

We have seen this technique successfully in use at one project. There, it was used as a bypass. Usually, the user gets a session ID cookie, but if their browsers do not accept this, they get redirected to another proxy in the DMZ that leverages SSL ID tracking.

If you intend to use this technique, make sure that you contact an experienced portal architect during planning. This has not been tested with WebSphere Portal and is therefore not supported. Additionally, we know that IBM GSKit, which is the SSL library used on the reverse caching proxy of IBM WebSphere Edge Server, and IBM HTTP Server only support a fixed number of SSL ID cache entries.

An often asked question regarding SSL usage is whether to use hardware SSL terminators or whether the HTTP server or proxy is able to do that. For the answer, evaluate what is possible with what costs in your configuration.

If a Web hosting company already has SSL terminators in place in your environment, they will likely want to push you toward this configuration. In such a case, there is usually good experience in quality and performance and they have worked out good ways for how to distribute the certificates securely.

However, SSL termination is a good chance to leverage the CPU power of the machines that you have in your DMZ. Understand that even if just a quarter of the responses are delivered SSL encrypted, it might lead to a double in CPU usage. Do not just go with this rough estimate, but make sure that you do proper load tests.

Note: Use the same topology and equal SSL terminators for your functional and non-functional tests as for your production environment.

In general, for SSL usage with WebSphere Portal, apply all the rules and best practices that also apply to any other multi-tier J2EE architecture.
Security beyond encryption

As we have learned in the introduction of this section, security is much more than just the encryption of data. Just to state two examples, part of your security includes:

– Ensuring that your portal does not break when a higher than expected load of users are accessing your system.

– Ensuring that the data in your database survives a hard disk head crash.

Although these situations appear to be logical for everybody, there are more delicate situations. Assume that you have an e-mail portlet on your portal and are writing a complex, time-consuming. When you click the Submit button, WebSphere Portal returns with the Login window and the message “Your session has been timed out. Please log in again.” What do you expect WebSphere Portal to do after login? Secure the user’s content and come back with the message “e-mail submitted successfully,” or secure the user’s account by ensuring that no tampered data gets into the account and always redirecting to the Welcome page after login.

What is your interpretation of security here? WebSphere Portal is able to do both, but you have to ensure that it is set up correctly to address your needs and that your application does not break the definition.

Non-technical parts of security concepts


Assume that we have portal system where all the best practices regarding password policies are checked and enforced. For example, the passwords need to consist of at least 10 characters including at least one number and one special character. The password must be renewed every three months, and the passwords must not be similar to one of the last six passwords chosen. If you forget your passwords, you must call a support line, a secure process. Because many more people than calculated forget their passwords, a lot of load was put on the support line, which raised costs and dramatically lowered user satisfaction. Therefore, the process was changed. Users then could state their user name to get their password from the operator. We hope to never see this type of security situation again.

> “Simply put, complexity is the worst enemy of security.”

Bruce Schneier, CTO Counterpane Internet Security, Inc.

Standard operating system level security considerations

There are a number of standard items that you should fulfill in order to raise the security on an operating system level. The following list mentions the most important:

– Do not run WebSphere Portal as the root user. Although it is quite reasonable to do the installation as the root user, in order to ensure that you do not run into problems, change the ownership of the installation directory after the installation and run WebSphere Portal under the account of a specially created system user.
– Ensure that developers do not have write access to the WebSphere Portal installation directories on any environment. They must not be allowed to add libraries or change configuration files on their own. However, also ensure that they do have read access. This will allow them to compare configuration files and, even more importantly, get an instant view on the log files. Not allowing them access often leads to waiting cycles and wasting time.

– Remove unused portlets and components. Unused components might unnecessarily extend the startup time, increase the memory usage, or negatively influence stability. Be careful, you must know what you are doing. If in doubt about a portlet or component, do not remove it.

– Do not use default passwords and establish a password expiration policy. We found systems on the Internet where you could log in with the user wpsadmin using the password wpsadmin. Even though these were only demonstration systems, make sure that this does not happen to you.

Single sign-on scenarios
As discussed earlier, single sign-on is one of the most discussed topics regarding portal systems, which is why we provide this section.

For more information while planning your portal systems security, see the “Security concepts” topic in the WebSphere Portal Information Center, available at:


The single sign-on “hype”
Almost every company runs a single sign-on project, sometimes called global sign-on. The original trigger for most is users being frustrated by numerous, different user IDs and passwords. For the most part, the federation of accounts is the goal.

Portal single sign-on realms
Occasionally, WebSphere Portal is assumed to be the single sign-on (SSO) solution or some other form of a virtual private network (VPN). It is neither. To understand what is meant by single sign-on, we first need to distinguish which SSO. Figure 2-8 on page 42 describes the SSO realms around WebSphere Portal. This figure shows a client Web application SSO and a portal back-end application SSO.

For additional information, refer to the IBM Redbook, Develop and Deploy a Secure Portal Solution Using WebSphere Portal V5 and Tivoli Access Manager V5.1, SG24-6325, available at:

http://www.redbooks.ibm.com/abstracts/sg246325.html

The white paper, Integrating WebSphere Portal with your security infrastructure, although a bit outdated, provides an excellent summary of the most important items of which you need to be aware, available at:

Here we introduce the previously mentioned SSO portal realms:

- **Client Web application SSO**

  At this layer, WebSphere Portal is often misused as a replacement to security systems such as IBM Tivoli Access Manager. WebSphere Portal does nothing else but leverage the functionality of WebSphere Application Server, which enables a single sign-on user experience with other applications that leverage IBM technologies, such other WebSphere Application Servers, Domino-based applications, or applications secured by Tivoli Access Manager. This is true if all of them leverage the LTPA token as an authentication token and use the same user directory.

- **Portal back-end application SSO**

  This is sometimes also called pseudo-SSO, because you are really not signing on with your single set of credentials. For example, you might have a portlet on your portal window that enables you to enter the credentials you always used for a certain traditional back end. The portlet saves the credential in a credential vault for you. Each time you access now your portal, it transparently logs you in to that configured system with the data out of the credential vault, transmitting, for example, Web services data from there and displaying it as you are used to on your portal window. This provides a big advantage, because the often difficult to understand authentication/authorization system of that established back end does not need to be touched. It is also often leveraged because it is an inexpensive way of migrating existing to portlets.

  Whether this is right for your portal system depends on your situation. Do not underestimate this technology. It has satisfied the needs of many customers. The cheapest, simplest approach is not always the worst.

---

**Figure 2-8  Portal single sign-on realm**

**Separate distinctive portals**

At this point, we want to mention the best practice that you should not mix systems with different targets. Assume that you intend to run a portal for your employees (B2E), for your customers (B2C), and your business partners (B2B). The intention is simple because there are some obvious consolidation effects. In particular, there is a chance of reducing costs. However, we do not recommend this approach, because of security and a possible distinctive growth and evolution.
Due to the goal of acquiring a single-sign on over such systems, we have never determined a reasonable business objective to pursue this. At IBM, we follow the same philosophy. An IBM employee has an intranet ID, with which that employee can access all relevant systems on the intranet. To retrieve an article from the IBM developerWorks® Web site, the employee logs in with his or her Internet ID. At that moment, the employee acts as customer/consumer. These IDs are saved in separate user directories.

### 2.6.5 Content management

Our experience with so many projects tells us that no matter what type of portal you are creating, it is likely that you will include some sort of content. This is a natural due to the importance of content in a portal project: Portals provide a single point-of-access that aggregates, personalizes, and delivers content and applications to users.

Content in a portal comes from a variety of sources: streaming video and audio, office-type documents, output from reports, and other digitalized assets. These files can reside in an already existing content management application, on a file share, or in a document management system. Integration with the portal can be as easy as downloading and installing a portlet from the catalog, for example there is a Domino Document Manager portlet for displaying those documents in WebSphere Portal. Or much more likely, it will be an add-on project that will take a significant amount of planning and implementation.

We have also observed the discovery of the need for a Web content management system. Many customers have a home-grown way to promote Web content to their Web site and now want to leverage IBM Workplace Web Content Management as part of their portal. A good way to analyze your content management requirements is to use the portlet sourcing exercise, described in detail in Appendix C, “Portlet sourcing” on page 105.

Discuss these points in your content and portlet sourcing meetings:

- What content do you want to display on WebSphere Portal?
- Where is the content stored?
- Who needs to create and change it, and who views it?
- What documents do you want to store?
- How many documents will be created each day?
- How large is each document?
- How many users will be accessing them (concurrently and total)?
- What will they be doing with them?
- What roles will they be assigned and how will workflow be used?
- What are the expectations regarding time to access, view, and store a document?

**Note:** Be sure to set realistic service level agreements regarding content presentation and manipulation.

The Content Repository API for Java Technology (JSR-170) specification will soon have a major impact on the content management landscape. It is a specification developed under the Java Community Process (JCP) with more than 60 members representing major content management solutions including Apache, IBM, Oracle, and BEA. JSR 170 specifies a standard API to access content repositories in Java 2 independent of the implementation. This specification continues to evolve and will specifically focus on getting content under control by offering a scalable and reliable infrastructure, something many customers are struggling with today as their repositories have spiraled out of control. JSR-170 will also attempt to offer friendly user interfaces, which promise a query service to search any compliant repository in a standardized fashion without learning a proprietary search API or language.
This standard offers great possibilities in a portal implementation effort. As a developer, this means that you will have one single API to work from without having to worry about which vendor’s repository is beneath the surface. As a chief information officer (CIO), you are probably managing several content management systems from multiple vendors. Having the same API on top of these existing applications will allow your organization to write applications quickly without having to duplicate application logic. As a Web editor, JSR-170 will allow in-place editing of content in the portal. Making changes in the directly in the portal, instead of dealing with a vendor’s interface, will reduce time to delivery and increase accuracy of data published on through your portal.

2.6.6 IBM Workplace Web Content Management

IBM Workplace Web Content Management is a key component of the IBM Workplace software solution, enabling users at all levels to manage end-to-end Web content creation and information life cycles, personalization, and publishing for dynamic intranet, extranet, Internet, and portal sites. Workplace Web Content Management allows the right information to get to the right people at the right time, so portals and Web sites are efficient and effective.

If you are a WebSphere Portal customer, you use the embedded version of Workplace Web Content Management V5.1. IBM Workplace Web Content Management is now fully integrated with WebSphere Portal V5.1. The installation of WebSphere Portal comes with a complete Workplace Web Content Management installation. Only configuration steps are now required to deploy Workplace Web Content Management. Workplace Web Content Management can be purchased and deployed separately or as part of a larger WebSphere Portal initiative. Refer to the IBM Workplace Web Content Management home page for more information:

http://www.ibm.com/software/workplace/webcontentmanagement

Workplace Web Content Management is easy to use although it can be cumbersome. Every hour of time spent on design will save days of rework. There are some critical aspects of design that need to be done up front. For example, it is important to discuss the content model. Determine what the site navigation will involve and what taxonomy is required. Workflow design and security should be part of this discussion. You will need to understand who can see the content, who can edit the content, and who can delete the content.

To be useful, content must be accurate and up-to-date:

- Ensuring that content is accurate and up-to-date is a big challenge.
- Content must be created, reviewed, edited, and approved before it can be published by the portal to end users.

Consider engaging IBM Services and begin your Workplace Web Content Management/WebSphere Portal initiative with a Workplace Web Content Management Architecture and Design Workshop. Figure 2-9 on page 45 shows the workshop objectives. This engagement provides you with the key steps to building a good Web site and offers hints and tips for using Workplace Web Content Management and WebSphere Portal.

Note: Today there is a gap between the WebSphere Portal content model and the site and site areas in IBM Workplace Web Content Management. This topic is addressed in Chapter 9 of the IBM Redbook, IBM Workplace Web Content Management for Portal 5.1 and IBM Workplace Web Content Management 2.5, SG24-6792. Therefore, we strongly encourage using this workshop to help you bridge that gap.
WCM Architecture and Design Workshop

- Describe the intended use of WCM.
- Demonstrate key milestones and checklists.
- Provide diagrams for use in WCM planning sessions.
- Include theory and practice behind a pragmatic approach.
- Deliver a reusable, step-by-step guide for planning and building new parts of your WCM solution.
- Help determine which tasks should be done to improve usability now and which tasks will be part of redesign.

Figure 2-9  Workplace Web Content Management Architecture and Design Workshop

We also encourage you to use the Techline sizing exercise from the Sales Productivity Center before you implement Workplace Web Content Management. This is a separate and distinct process and uses a separate checklist.

Content delivery

One of the most important issues you need to analyze is where you will deliver your content:
- Stand-alone Web site (through Workplace Web Content Management rendering server)
- WebSphere Portal (through rendering portlets or custom portlets)
- Stand-alone Web site and WebSphere Portal

Roles

In addition to the project team listed earlier in 2.2, “Roles and responsibilities” on page 16, you would typically see the following roles in a Workplace Web Content Management/WebSphere Portal project. These individuals make up an information architecture team. This team should have representatives from all content areas, for example, human resources. It is important to have this team led by a single person who is authorized to make decisions.

- Designers
  Designers help to create the corporate design. They work with the IT team to technically implement the corporate design. Designers also help to create the authors’ template environment. This is the way content will be added into the environment.

- Subject matter experts (SMEs) and authors
  SMEs and authors provide the content. They help to determine the taxonomy and content structure. They should also have input to the design of templates and offer support for them in a deployed environment.

- Content owners
  Content owners are typically content authors as well. They strive to provide quality assurance of the content. Content owners participate in pilots are provide valuable usability feedback.
Target group (end users)
End users will help to outline their business needs and provide you with feedback during a pilot.

Architecture
There are several ways to architect a Workplace Web Content Management (Figure 2-10) environment. Take careful consideration when designing your implementation. Each customer’s needs are different and there is no best practice guide. We strongly recommend the consulting services of an IT architect before you begin to configure your environment.

The following numbers refer to Figure 2-10:
1. Remote rendering in a portal environment using the remote rendering portlet to show published content on a different server.
2. Non-portal delivery of Workplace Web Content Management content. Note that the Workplace Web Content Management rendering server is a Workplace Web Content Management module that needs to be is installed in the Portal server, but portal delivery is not required.
3. Portal delivery within Workplace Web Content Management environment using the local rendering portlet to show published content on the same server the authoring takes place.
4. Authoring UI in WebSphere Portal. The authoring portlet is used to create and manage content.
**Restriction:** The authoring portlet does not function properly on a Workplace Web Content Management server that is installed into a cluster. Authoring cannot be done in a clustered environment. Furthermore, in order to make configuration changes to production servers (some of which are done through the authoring portlet), you have to remove the server from the cluster prior to making such changes.

For more information about architecture, consult the IBM Redbook *IBM Workplace Web Content Management for Portal 5.1 and IBM Workplace Web Content Management 2.5, SG24-6792*, available at:

http://www.redbooks.ibm.com/abstracts/sg246792.html

**Restriction:** The current Workplace Web Content Management software cannot take advantage of clustering with respect to the content repository. Each server that has Workplace Web Content Management enabled will need to be configured to use a separate database. This is the exact opposite of the WebSphere Portal database, which must be shared by all portal servers in a cluster.

**Your Workplace Web Content Management infrastructure**

In a typical IBM Workplace Web Content Management installation, there are multiple physical Web Content Management servers. In our experience, these servers end up performing one of four roles within your IBM Workplace Web Content Management infrastructure:

- Development
- Authoring
- Staging
- Production (live)

Development is where you create and unit test the Workplace Web Content Management technical assets such as presentation templates, HTML components, menus, and navigators. In addition, this server can be the first place you install patches and fix packs to ensure that they do not have a negative impact on your Workplace Web Content Management servers.

Typically, you syndicate changes to the authoring environment to roll out changes. These changes are, in turn, syndicated or replicated out to the staging and live environments after the appropriate testing.

**Restriction:** Workplace Web Content Management does not currently support the notion of selective replication. In other words, you cannot tell Workplace Web Content Management to only syndicate design changes from the development server to other servers in your environment. This limitation means that you must be very careful when syndicating content to and from a development server.

One approach is to set the syndicator to syndicate only live content and delete all content (through the API) prior to syndication.

Another similar approach is to configure syndication the same way but have no live content in the development server.

Regardless of your preferred approach, be very careful when syndicating from the development server to other servers.
Realistically, there are a variety of common infrastructure designs in place. The primary differences in these designs are due to variations in several basic assumptions:

- Site size and complexity: For a small Web site, it might not be necessary to have individual servers dedicated to all four types of Workplace Web Content Management environments.
- Funding: In many cases, there is a limitation in the funding provided for building out the Workplace Web Content Management infrastructure.
- Corporate standards: It is quite common for smaller companies to omit the staging environment because it requires time and resources to perform the content review in this stage.

Refer to *IBM Workplace Web Content Management for Portal 5.1 and IBM Workplace Web Content Management 2.5*, SG24-6792, for a sample of a basic, intermediate, and advanced sample architecture. There is no single answer that will work in all scenarios. The IBM Redbook includes key success factors such as designing a user-centric Web site and planning the site framework and site areas. Taxonomy, metadata, and workflow are other topics for consideration and are discussed in great detail.

**Cache strategy**

A typical IBM Workplace Web Content Management implementation shows dynamically generated pages, often combining navigation elements with files (such as style sheets or JavaScript scripts), images, and site content.

Because Workplace Web Content Management is run as part of a WebSphere Portal environment, Workplace Web Content Management performance depends on many settings outside of the Workplace Web Content Management configuration. For example, Workplace Web Content Management performance depends on hardware, Web server caching, WebSphere Application Server settings, and WebSphere Portal server settings.

For more information about architecture, consult Chapter 8 of *IBM Workplace Web Content Management for Portal 5.1 and IBM Workplace Web Content Management 2.5*, SG24-6792, for strategies to improve performance for:

- Caching
- Pre-rendering
- Dynamic cache

Dynamic cache (WebSphere Dynamic Cache Service) is a WebSphere service that is enabled on an application server by default. It supports caching of servlet and JSP responses, WebSphere commands objects, Web services objects, and Java objects.

**Important:** Before implementing or testing a caching strategy, it is important to be aware of other caching systems that might affect Workplace Web Content Management performance. If Workplace Web Content Management is delivering content through WebSphere Portal, consult the portal administrator and determine if portal pages delivering Workplace Web Content Management content are configure to be cached by the portal. We discuss portal's caching system in greater detail in Chapter 3, “Developing a portal” on page 59. In addition to portal's cache, review the cache settings on any Web servers delivering Workplace Web Content Management content configured with the WebSphere Application Server plug-in. Basic caching should not be affected by the Web server's cache, but if an advanced cache strategy is used, ensure that the Web server's cache settings are configured to deliver secure and personalized content. Otherwise, performance might degrade because your Web server is caching too much or constantly rebuilding the cache.
A good paper to read about this topic is *Using WebSphere Dynamic Cache Service with IBM Workplace Web Content Management*, available at:


**Search strategy**

If your site is large or contains a wide variety of content, site visitors will expect to be provided with one or more techniques for searching the site for content. Workplace Web Content Management V5.1 provides you with multiple techniques for searching its content repository:

- Workplace Web Content Management integrated search module
- Workplace Web Content Management API
- WebSphere Portal search
- Third-party search product (for example, OmniFind™, Verity, Lucene)

Each search option has its advantages and disadvantages. In many situations, you will need a combination of these capabilities to meet your specific requirements.

We encourage you to review Chapter 10 of *IBM Workplace Web Content Management for Portal 5.1 and IBM Workplace Web Content Management 2.5*, SG24-6792, for more information regarding searching. You should determine in the planning phase which strategies you will incorporate in your Workplace Web Content Management/WebSphere Portal project. The security settings on content are taken into account during a search. Searching is performed on the content objects, keywords, and categories stored in the repository. A good deal of forethought about content creation and security must be done in order to retrieve reasonable search results.

**Note:** The Portal Search Engine has a spider that crawls WebSphere Portal sites and regular Web sites. There is a difference between the way the crawler follows the links inside the sites. A way to overcome this is to build a custom portlet. The Workplace Web Content Management API provides you with access. Refer to Chapter 10 in *IBM Workplace Web Content Management for Portal 5.1 and IBM Workplace Web Content Management 2.5*, SG24-6792, for instructions about building this portlet.

**Migration strategy**

Undertaking a migration of Web pages or documents into a content management system (CMS) is often approached as a “last minute” exercise. Normally, assurances from CMS vendors imply that there are no issues associated with this part of implementing a successful CMS strategy. However, like many other aspects of IT implementations, it is the correctness and validity of the data entered into a system that determines its success or failure. This is why IBM has teamed with Vamosa to recommend the use of Vamosa's unique toolset to migrate all of an organization’s existing content into Workplace Web Content Management.

Note that Vamosa’s technology represents one of the strategic options for migrating content into Workplace Web Content Management. There are other technical approaches available. Regardless of which specific technology is used to migrate content, you must determine the business case for a migration.

It is important to clearly define your measurement of success for the migration effort.

Based on the constraints placed on the project, it is necessary to identify and ratify the approach and required toolset necessary to satisfy the objectives. For example, if the volume of data to be migrated is significant, an automated approach is the best solution. For smaller volumes, usually less than 5000 pages, a manual approach should be sufficient.
The actual undertaking of the migration should be a relatively straightforward task. All of the previous definitions have set out a framework of operation that reflects both source information and the required target system. By defining both the as-is and the to-be requirements, as well as the transformation rules and exception handling, combined with the QA necessary for the migration, all the technical elements are in place to proceed.

Although the theory of achieving 100% data migration is the goal of any migration, it is usual in most circumstances that data is migrated within the context of iterative migration sets that work toward a 100% success rate. Typically, the “business rules” that are designed and defined will exploit a large subset of the data to be migrated. However, after the first set of runs, there will be a minor subset of data that has not been addressed correctly. Additional “business rules” will be defined and the migration steps will be rerun. This process will be repeated until a high enough success rate has been achieved.

2.6.7 Search

We provide this section to help you understand the general process of building a search index or search collection. We also describe details of typical problem areas that you might encounter.

The WebSphere Portal Information Center devotes an entire chapter to the topic of WebSphere Portal Search. Nonetheless, there are recurring questions regarding this topic.

As customers acknowledge the need and importance of text search capabilities in WebSphere Portal, they also find that creating search collections is not as easy as originally anticipated. Though this does hold true for most of the cases, we do see customers challenged with situations where they do not understand what WebSphere Portal Search is doing and what is being crawled and need help about how to proceed.

We divide this section into two parts: The first part provides an overview of the WebSphere Portal Search Engine itself and then another that covers frequently overlooked topics. IBM development continues to provide enhancements with every release, soon perhaps rendering this part of the paper obsolete. Until that time, this should provide you with good guidelines about how best to use Portal Search.

Because this section is based on the ideas of Andreas Prokoph’s paper A Guide to using WebSphere Portal Search- First steps, you might also want to refer to his up-to-date version on the Web at:


For a clustered environment setup, we recommend reading Setting up Portal Search in a WebSphere Portal V5.1 clustered environment, on the developerWorks site at:


Overview of the Portal Search Engine

WebSphere Portal provides integrated portal site and Web content indexing and search capabilities. In addition, there are advanced features such as methods for content categorization of indexed content and an optional workflow approvals process to manage the publication of indexed content. There are two search portlets provided for the end user, one that offers a basic Web-style search and another that offers the ability to perform an advanced search.
To implement the search capability in your portal, it is important to understand the basic mechanisms of how information and data flows from content sources until they are stored in the full-text indexes. It might be also interesting to note here that the way WebSphere Portal Search works applies to most of the Web search engines available today.

Figure 2-11 shows the flow of data (content) and the major components involved.

![Figure 2-11 The flow of data (content) and the major components involved](image)

In order to fetch and store content in the search index, the following processing steps apply:

1. The **crawler** fetches content from a specified location, which is typically a starting point URL. For Web content, it is assumed that all relevant information can be gathered by following the hyperlinks in the text.

2. For every **page**, the crawler consults the rule sets or **filters**, if available, to determine whether or not this new page is important for further processing or if it is to be bypassed.

3. If the page passes the filter criteria, it is sent for processing to the **text analytic components**. Here, the page is checked for **incoming information** such that it can be efficiently processed and stored in the full-text index.

4. All required information is now available to store the page information in the search index. The last processing step is then performed by the **indexer** whose task is to merge the information about the page into the new or existing search index.

**Important topics while working with Portal Search Engine**

For more details regarding Portal Search, see the Information Center, which dedicates a chapter to the Portal Search Engine, available at:


The following sections describe some topics that are of major importance when implementing Portal Search Engine.

**Completeness**

Ensure that the crawler can technically reach all content to which users typically have access. If a single entry point for the crawler is not enough, adding more **content sources** or defining a
seed list, for example, a page with a list of URLs, and pointing the crawler to that list will certainly help.

Ensure that all content that it is hyperlinked through the home page or the initial URL provided to the crawler can be reached. Many content management systems provide integrated content in a Web site and ensure proper linkage among the pages. Many systems will allow you to generate a site map that can be provided to crawlers. A good starting point that will finally lead to complete Web site coverage is essential.

Note if there are too few pages crawled and indexed. The administrator in charge of setting up the search collection should have a good feeling about how many pages there are. If the two numbers do not match, the crawler might not have the full access.

There is no perfect checklist about how to build a good site map or a seed list. This again is an iterative process. Include a person within the organization who is responsible for taxonomy as part of your planning and review process.

There is a non-interactive command line tool called *wget* that is worth investigating. It is free of charge, because it is provided through the GNU Open Source License. If you are in doubt as the validity of your collection coverage, try using this tool to explore the site structure. You might also include it in a regularly run UNIX cron job, for example, to check for changes in certain site map branches. For more information, see:

http://www.gnu.org/software/wget/wget.html

**Crawlability**

Crawlers, by definition, behave somewhat differently than users with a Web browser. In the past, crawlers did not honor cookie requests. The main reason was that cookies are often used for personalizing a Web site experience and crawlers assumed *anonymous* access to Web sites. This was true until WebSphere Portal Version V5.0.x. Beginning with V5.1, the WebSphere Portal Search Engine crawler began to offer support for cookies. The biggest hurdle yet remains with JavaScript. Crawlers do not interpret JavaScript at this time.

Therefore, the crawlability of a Web site depends on multiple factors. The most prominent are:

- **JavaScript**
  Crawlers do not provide a JavaScript interpreter of their own. One of the reasons is again because of its use for personalization and security. Therefore, the crawler will simply skip JavaScript clauses found in the HTML sources. One could argue that URLs are often seen in the JavaScript itself, and a simple parser could help in pulling them out and using them as a page reference. However, in those cases, it is unknown as to whether the crawler is eligible to do so. An example is an “if-else” clause in JavaScript that provides a link for the anonymous user and the other link for the authenticated user. Although this is not a good practice in terms of security, JavaScript is often used this way.

- **robots.txt**
  The search engine looks for a special file named robots.txt. The robot directives are usually defined and configured by a Web site owner rather than by a portal site owner. If the crawler is prohibited from crawling the site or parts of the site, this request must be honored; even the administrative interface of the crawler allows the crawler to ignore the robots directives. If you really want to use this feature, which is mostly only required within a company's intranet, it is a good practice to inform the affected Web master or site administrator and request permission to crawl those areas.

Here are a couple of general reasons for setting up the robots.txt directive file:

- Enforcing *copyrights* on published materials.
– Locking out crawlers from visiting certain areas of the Web site, for example, preventing the submission of an order forms or initiating other types of actions.

– Load balancing. There are many crawlers active on the Internet. These can potentially hit a Web site one or more times per day, affecting performance. To protect the user experience of a Web site, many crawlers will be prohibited from visiting the site. Note that crawlers generate a much higher usage rate that typical users do. Consequently, only the very popular ones will be allowed to run.

The robots.txt file applies to Web sites only. Such a mechanism is usually not useful for portal sites. This is because the robots.txt file relies on patterns within URLs to be descriptive and used as a unique reference to pages or subdomains of a site. However, in WebSphere Portal, the URLs are of a more dynamic nature and thus render them unusable.

▶ Site maps

Take advantage of a content management environment if available. These are typically able to create a site map for you on the first page, making crawling very efficient.

– A site map should include all pages that a crawler should fetch.

– It should not contain more than 100 to 200 links on a single page. If more links are available, place a “next page” link at the end and point to the continuation page.

– If placed on the first page, the crawler can efficiently traverse through the list of links. If it encounters additional links off visited pages, it then determines if it has already visited that page. The efficiency of this really depends on the crawler’s implemented logic.

– The site map is an option used to determine a list of all pages to be crawled and indexed. This can also be provided by the home page if you are certain that the crawler can reach all subsequent pages. We have seen too many examples where this was not the case. An extreme example is the use of an initial Macromedia Flash splash screen and not providing a crawler friendly way to bypass the initial screen.

Note if the crawler starts, but comes to an end after a few minutes. You might have run into one of the cases where crawlability failed, perhaps due to JavaScript, robot directives, or a home page that does offer full coverage.

Note: If the WebSphere Portal Search Engine crawler is not able to crawl your site, the crawlers of Internet search engines such as Yahoo! and Google will not either.

Configuration and administration of search collections

Search collections are partitions of information logically grouped into independent content sources. The goal is to ease overall search performance, thereby increasing the quality of the search results. Part of the partitioning can be performed by referencing subdomains of the overall Web site, others by applying filters that define what content the crawler fetches and whether or not it is then also indexed.

A sample partitioning for a search collections might be:

▶ Regular content

Content that is updated frequently should be defined as a single content source. In many cases, the content is published as HTML. Here, it might be, for example, all standard HTML pages with a decent size of a 10 KB maximum.

▶ Static content

This might be for archived materials. The update frequency can be adjusted to once every quarter or less often, for example.
The processing of large items such as PDF and zipped files can take a considerable amount of time. It is important to strike a balance about what to index and how often.

The following steps describe how to create a search collection:

1. Create the search collection.
   Consider the delta between features offered and the performance impact they might create. By using the non-default, advanced features, the processing time might increase from insignificant, for example, about 30%, to very significant. The factors relate to a number of parameters, such as pages per document, size of text, and size of vocabulary.
   However, users might appreciate certain features of the Portal Search Engine such as the summarizer.
   Therefore, you might not use every feature, but instead, do performance tests or at least do some calculations on it before you promise nice-to-have features.

2. Define one or more content sources.
   For each group of a search collection, you will need to define a single content source entry and provide the adequate definitions and configuration to ensure that the best possible search collection is built.
   One of the definitions that you might want to adjust is the number of crawler threads. This can save system resources.

3. Initiate the crawler process.
   A frequently asked question is: “It seems that the crawler always runs for hours even though it does not really look like that it is doing anything because the hard disk does not seem to be in use. Is this normal?” You might have run into the problem of not adequately separating your search collections. Proper filters might not have been applied, and the crawler might have run into GB-sized ZIP files, so now the document converter is trying to generate an HTML representation from it.

In summary, the advantages you gain from partitioning your search collection include:

- Throughput of data significantly increases. Your focus should be on content that typically changes often.
- Static content, parts of your site that are not updated very often, can be set up to crawl with very long intervals. This saves system resources.
- Processing of large documents can be isolated within a separate process that is invoked (scheduled) independent of the more rapidly changing content.

**Tip:** It is worth the investment of your time to plan for search collection partitioning.

**Search collections in a cluster scenario**

WebSphere Portal Search Engine builds its search indexes on the file system of the server in V5.1. Therefore, there is no way of clustering this. You can install Portal Search Engine on one of your portal servers and then from within the cluster, configure WebSphere Portal to use that search service, thereby creating a remote search function. This does mean that you have a single point of failure as far as search is concerned. In the future, Portal Search Engine will build its indexes in the database and they will become part of the portal cluster.

For more information, consult the WebSphere Portal Information Center:

It is very important to plan your search collections before you start the collection process. The portal site default search collection is created only once. This happens when the portal administrator selects the search administration portlet, Manage Search Collections. If this is done before you configure the portlet for remote search, the default portal site search collection is only available on the primary node of the cluster and is not available on the remote server. If this happens, you need to re-create the portal site collection to make it available for search on all nodes of the cluster.

2.6.8 Virtual portals

Virtual portals are a feature that became available with Version V5. Best practices for tested and implemented virtual portals are not possible at the time of this writing. It might be, however, a good topic for a future paper because most customers are anxious to deploy production portals in this way. The following discussion about history, concepts, and the distinction with true portals should help you determine whether this technology is of help for your portal project.

The concept of virtual portals and their implementation

Figure 2-12 illustrates the key items on which the concept of virtual portals is based.
The following numbers refer to Figure 2-12 on page 55:

1. A virtual portal is accessed through a specific URL mapping, which is associated with that virtual portal. By definition, each virtual portal must be accessed with a unique URI; WebSphere Portal takes care that the requests get dispatched accordingly.

2. Virtual portals can be customized to expose a unique look-and-feel. Different themes and skins can be assigned to different virtual portals. In contrast to true portals, you must not use screens but now pages (see 1.6, “Layout of a typical WebSphere Portal implementation” on page 7 for a definition) to implement your login, for example. Due to the implementation of screens, they are a concept no longer usable for virtual portals.

3. The notion of scoped and shared portal resources is introduced, which allows isolation between the content of different virtual portals. This was introduced to the internal WebSphere Portal data model starting with Version 5.

4. Each virtual portal can have its own distinct user population. This was originally solved by grouping users of a single user directory. Now, we can also address various directories.

5. For the administration of virtual portals, the delegation model of WebSphere Portal access control is leveraged. Within a virtual portal, subadministrators can apply access control independently. The technique of subadministrators had always been available in WebSphere Portal. The feature of virtual portal required additional portlets to leverage it.

6. Login and Selfcare are implemented as a portlet and can be customized for a specific virtual portal. As already described, the implementation concept of screens did not allow a transformation to virtual portals. Therefore, the functionality formerly only available as screens was moved to portlets, which helped more than the virtual portals feature. For example, the management of various versions is easier with portlets than with screens.

7. A new administration portlet exposes a user interface to manage virtual portals.

For more information about virtual portlets, refer to:

- The WebSphere Portal Information Center, available at: 

- Articles on developerWorks, for example, “Creating portal instances on demand,” available at: 

The difference between true multiple portals and multiple virtual portals

If virtual portals seem to fit for your project, carefully plan and consider whether this is the right technology, or if it is an advantage to run true portals.

WebSphere Application Server 4.x introduces the concept of true multiple portals, because it was then possible to install the software multiple times on a single environment. Therefore, in contrast to running multiple virtual portals, you share less resources but get full individual configuration options. Table 2-1 on page 57 gives an overview of what resources are shared in which configurations. This table also includes the column “Virtualization features,” which might be the IBM AIX® 5L™ feature of LPARs or VMware installations. Understand that we do not recommend VMware-based portals for production usage. IBM will not support a portal implementation on VMware. We see customers using this platform to save costs for some environments, for example, test environments.
Table 2-1  Description of the level of sharing

<table>
<thead>
<tr>
<th>Resources</th>
<th>Virtualization features</th>
<th>True multiple portals</th>
<th>Multiple virtual portals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical hardware resources</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Operating system</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>WebSphere Application Server and WebSphere Portal libraries and settings</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Important:** The more resources you share, the more dependencies you create.

Due to the steadily increasing pressure to reduce costs, the pressure to share or reuse components and resources increases as well.

Assume that an environment gets only load once a month or once a quarter. You want to find ways to leverage this idle hardware the rest of the time, but you will likely need to act strategically and thoughtfully, because it might be of high importance that this environment performs very well that single day. Some new IBM AIX 5L features might be of help here.

Similarly, you can save costs by hosting multiple application on a single server. Just combining only two applications on a single operating system saves you the administration and maintenance of one operating system. If there critical OS updates available, for example, security fixes, you need to upgrade one less system. Assume, however, that the project iteration just finished and they intend to upgrade the application. It might now have different prerequisites, such as requiring a later OS level. If the other applications are not compatible with an upgrade, you need to separate them again. Additionally, can you ensure that none of the applications will break the OS and therefore stop all other applications?

**Note:** By sharing resources and thus saving costs, you buy in risks.

We agree that the technologies discussed have been used for many years and there are operating systems with very good tools that ensure that certain applications get their allocated CPU time and use of memory within an established maximum.

At the level of virtual portals, there are many technology dependencies between the portals, and we cannot assume that there are proven tools that ensure that the portals cannot damage each other.

In summary, we conclude that virtual portals are a great technology, if you have the right use for them. For example:

- Portals that have the same development iteration cycle and the same development team
- Portals that use the same applications, maybe even the same user groups and have the same project owner
- Portals that have similar usage targets, thus requiring equal security considerations

Some of these arguments might fit into your portal systems; therefore, the question is whether you can take the risk. At single portal systems, we often see problems because of class loading, for example. A new, correct portlet gets deployed and problems still occur because of some previous mistake that did not yet show. We often see people with projects where they must look for memory leaks within one of the portlets or the deployed portal add-ons. If you
increase the set of code and the number of people involved that own code on the portal, these problems might multiply. In addition, note that logging is not scoped to the virtual portals, because there is only one WebSphere Portal installation based on one WebSphere Application Server installation.

From a project owner point of view, you will understand that your portal system is not up and running if some code in that system does not behave correctly. However, you might have a hard time understanding this if it is because of the code of some other project.

If you need to host more than one portal system and have the requirement to share resources such as hardware and OS, also consider installing WebSphere Portal multiple times in your environment. The cost factor that increases here is not CPU, and often it is also not administration. Today, because port conflicts are not an issue any more, the additional administration for the additional application server out weighs the problems that you might encounter by hosting the virtual portals of separate portal project development teams. However, your requirements regarding RAM will be much higher.

**Tip:** WebSphere Portal licensing does not distinguish between true multiple portals or multiple virtual portals. You are allowed to install as many portals on a machine and as many virtual portals.

The interesting factor for licensing is the number of CPUs you are leveraging based on the pricing model.
Developing a portal

A well-designed portal can provide a common user interface and content base that can be integrated and leveraged across all portal applications to deliver a unified, collaborative workplace. In this chapter, we discuss design, integration, and performance topics to help you create a first-class portal application.
3.1 Customization versus advanced personalization

Before we talk about developing a portal application, we need to discuss some terms that will help you understand how you can create a unique experience for your users. We use the term customization to mean the rendering of portlet content based on users’ preferences or manipulating the portal layout based on the users’ security attributes. We use personalization to mean delivering portlet content based on a business rule or collaborative filtering. Collaborative filtering uses statistical techniques to identify groups of users with similar interests or behaviors. Inferences can be made about what a particular user might be interested in, based on the interests of the other members of the group.

Customization centers around what resources (portlets and pages) you show the users based on their role. This is a core capability of WebSphere Portal. There are tools provided that help administrators provision these resources. Typically, portlets enable users to specify what kind of information should display. For example, a weather portlet might be customized to show weather in the user’s home town.

Personalization can deliver content to a user based on a profile and business rule, or determine characteristics of a user based on previous purchases or pages viewed. Personalization then selects content that is appropriate for that profile. For example, if a person has a high salary range, personalization can be configured to retrieve information about premium products; the page is assembled with the proper information and the user sees the personalized page.

Include some planning time for deploying personalization so that you can optimize performance. Personalization includes a rules engine, a recommendation engine, and a resource engine. Although a full installation of WebSphere Portal installs and configures personalization functions by default, there are additional databases to consider. For example, personalization uses IBM DB2 Content Manager Runtime Edition for the storage of rules, campaigns, and other objects. There is a logging framework used to record information about Web site usage to a feedback database. The LikeMinds recommendation engine also requires a database to store information gathered through the logging APIs.

To use personalization, you need content to display to your users. Therefore, personalization and content management go together. You will also need to understand where content is stored and how it is managed to optimize performance.

The Techline sizing exercise does not take into account the additional resources (processor, memory, hard disk, and so on) that you might require for using personalization yet. The performance lab will hopefully include this in their metrics in the future. However, for the moment, we do not have any good quantifiable rules to apply here.

3.2 Developing portlets

IBM Rational Application Developer enables you to quickly design, develop, analyze, test, profile, and deploy Web, Web services, Java, J2EE, and portal applications within its integrated development environment (IDE). Rational Application Developer and the Portal Toolkit are tightly integrated with WebSphere Portal. You will receive one license of Rational Application Developer for every WebSphere Portal server you purchase. Rational Application Developer generates the model view control (MVC)-compliant code for both the IBM and the JSR-168 portlet API. See 3.4.1, “Caching” on page 68 for more information about these two. The IBM Redbook IBM Rational Application Developer V6 Portlet Application Development and Portal Tools, SG24-6681, will help you get started with development:

http://www.redbooks.ibm.com/abstracts/sg246681.html
An excellent resource for developers is the IBM developerWorks Web site, provided by experts at IBM to assist software developers. developerWorks is a place to access tools, code, training, forums, standards, and how-to documentation. Refer to the following link:

http://www.ibm.com/developerworks/

As you begin to develop your own portlets, you will create many versions of the same portlet. Consider these portlets as assets and collect them as you would tools in a toolbox. Try to keep your portlets simple. Designing portlets to be flexible, providing the end user with many configuration options, is risky. This often confuses the end user and leads to unexpected results on the portal page. Consider creating multiple, single-purpose portlets instead.

**Tip:** Avoid creating the super-configurable, multipurpose portlet. Your development goal is to keep portlets small and simple, making them easier for other developers to understand and to maintain.

### 3.2.1 APIs and frameworks

There are two portlet specifications available today. Which one should you use? One portlet container is for the new JSR 168 Portlet API and the other is for the IBM Portlet API that WebSphere Portal supported before JSR 168 was available. When creating a portlet, a developer has to declare which type to use. The IBM Portlet API offers more functionality than the Java portlet standard. For example, the IBM Portlet API makes better use of native WebSphere services such as retrieving authentication credentials. However, portlets built to the IBM specification cannot be used in any other portal, only WebSphere. IBM has indicated that it will eventually end support for the IBM Portlet API at some point in the future. It is not imminent, however, and it is possible to migrate from the IBM standard to the Java standard.

If you have a need for one of the Domino portlets, there is no choice; they support the IBM API. However, in WebSphere Portal V5.1, IBM portlets can share data with JSR 168, so you can create new portlets to interact with them.

**Tip:** Use the JSR 168 Portlet API whenever possible.

IBM strategic direction is based on the JSR 168 standard. For developers, this means that IBM portlets will most likely have to be migrated to JSR 168 at some point. In addition, JSR 168 will probably get better with time as new versions come out with added capabilities. In the meantime, IBM provides extensions to enable JSR 168 portlets to take advantage of some of the capabilities of WebSphere, according to Stefan Hepper, architect for the WebSphere Portal Server Development at IBM. Read his paper, *Portlet API comparison white paper: JSR 168 Java Portlet Specification compared to the IBM Portlet API*, available at:


Struts is a very popular framework for Web applications using a model view controller design pattern. The Struts framework can be used to effectively design Web applications and support development teams of different sizes and organizations. It is important to note that WebSphere Portal V5.1 provides a Struts portlet framework to be used with JSR 168 portlets. A good example of where to use Struts is for portlets that require a wizard interface.

JavaServer Faces (JSF) is a technology defined by the JSR 127 standard that helps you build user interfaces for dynamic Web applications that run on the server. The JSF framework manages UI states across server requests and offers a simple model for the development of server-side events that are activated by the client. WebSphere Portal V5.1 includes support
for JSF portlet applications by providing a JSF portlet run time that makes it possible to run
JSF applications as portlets in WebSphere Portal.

JSF will dramatically change portlet development in the future. It is important to start learning
JSF. Refer to the following excellent sources of information:

- The WebSphere Portal V5.1 Information Center:
  http://publib.boulder.ibm.com/infocenter/wpdoc/v510/index.jsp
- An article about developing JSF portlets with Rational Application Developer:
  http://www.ibm.com/developerworks/rational/library/05/genkin/

### 3.2.2 User interface design

Successful project implementation begins with setting proper expectations. Be careful what
you agree to regarding customizing the portal theme. We have often seen where an
organization will hire a marketing firm to create a new image or presence for them on the
Web. This engagement results in the delivery of a mock-up Web site in the form of a JPEG
file. This file is subsequently handed off to the portal implementation team, sometimes well
into the project, with the direction, “Our portal needs to look like this.”

Understand that customizing a portal theme to match an exotic, elaborate, or complex Web
site can be very time-consuming.

Over and over, we have seen teams agree to create a custom theme without performing the
due diligence necessary to include a reasonable time estimate in the project plan. We have
seen project deadlines slip because this step was grossly underestimated. One way to
mitigate risk here is to settle on the theme of the portal before the project begins. Another way
is to educate your user interface design team about portal capabilities.

**Tip:** Designers who understand core functionality of portal are more likely to design an
interface to leverage those capabilities.

There are several approaches to creating a custom theme. If you are new to theme building,
and the theme is not too complex, try the IBM WebSphere Portal Theme Builder portlet. This
might already give you a good starting point. You can download this portlet from the IBM
Workplace Solutions Catalog:

http://catalog.lotus.com/wps/portal/portal

This is one of the most popular downloads and enables you to create new themes with
customized basic branding. The portlet provides a preview window that shows you what the
current theme will look like. This approach is good for those who do not have any HTML skills.

A second approach is to copy an existing theme and modify it. This is a good idea if you are
familiar with HTML and cascading style sheets. The basic installation of WebSphere Portal
provides you with several themes. You can review the sample themes and select one that is
close in design to the one you are working toward. You can use any tool with which you are
comfortable, but note that Rational Application Developer has a built-in theme and skin
designer. Here is a good article that describes, and provides as a download, several
examples of themes and skins:


Again, be careful what you agree to design. Carefully manage the amount of content and
complexity of the layout, especially if you are new to WebSphere Portal. On your first theme,
use only a single style sheet (styles.css). Using more than three or four navigation levels is not advised, because it becomes difficult to manage. Another tip is to be realistic about the numbers of portlets users have access to on one page. This can result in a performance problem. Also, for better manageability, we suggest that you avoid placing portlets in rows; use the column containers. Another best practice is to add lightweight portlets on pages everyone accesses, such as a bookmark portlet. Add more complex portlets to pages that users select, for example, the mail portlet.

No matter which approach you take to design a custom theme, get end users interacting with it as quickly as possible. Do not wait until the end of your project to get feedback. Expect many iterative changes in this process.

### 3.2.3 Markup generation

No matter how well you design and implement your portal project, it is all worth nothing if the output it delivers is unsatisfying.

**Pervasive access**

WebSphere Portal was originally designed as a pervasive portal by IBM Pervasive Computing Development. Therefore, its major goal was to have a portal that can be accessed from anywhere and from any device. Therefore, the portlets have the possibility to deliver different content for different markups and the portal has to describe which markups it is able to deliver in the portlet descriptor. Based on this information, WebSphere Portal will exclude certain portlets if they report being unable to deliver a certain markup.

This technique allows an easy implementation of pervasive portals, portals that support a broad range of markups and therefore devices. You can design and create a portal for HTML access and have at the same portal a user-interface optimized for Wireless Markup Language (WML) access to be leveraged by Wireless Application Protocol (WAP) clients.

Over the years, it turned out that the usage scenarios of, for example, an HTML and a WML markup-based portal do sometimes differ significantly. Therefore, products such as WebSphere Everyplace® Access and WebSphere Everyplace Mobile Portal were delivered. These products leverage WebSphere Portal as a core engine but add additional features on top of it. WebSphere Everyplace Mobile Portal, for example, extends the WebSphere Portal server to meet the requirements of mobile and wireless service providers. WebSphere Everyplace Mobile Portal also introduces an additional markup that enables developers to create applications independent of the specific devices, but just specific to certain broad categories. Here, developers create a meta-markup called XDIME, which stands for XML Device Independent Markup Extensions. This allows markup to be generated once without the author having to know what specific device it will be delivered to and thus addresses the very short life cycles on the mobile device market. Based on a constantly refreshing database, WebSphere Everyplace Mobile Portal decides how to generate the appropriate WML or XHTML markup that gets delivered to the real device.

Still some corner specifications have big advantages or are necessary for development. These include screen size, type of input device, connection speed, and computing power. To describe an extreme, it is obviously different if you design an application of an IBM WorkPad c3 PDA with an 160x160 pixel black and white screen that comes with a Palm Modem that can reach a maximum of 33.6 Kbps than if you design an application for a Sharp Zaurus SL-6000 PDA that comes with an 480x640 pixel color screen, integrated WLAN, and even a small, integrated keyboard. Therefore, understand that you can deliver the markup for certain devices; you have to ensure that the applications are also usable on those devices.
Note: We believe that portals with a poorly designed user interface and applications that do not leverage the mobile devices capabilities are the main factor of why there are not a lot of successful pervasive portal systems in production.

Being able to address the expectations users who want to work with their specific devices will be a key factor for a pervasive portal project.

For more information, consult the following articles about how to use WebSphere Everyplace Mobile Portal:


For more information about WebSphere Everyplace Access, consult the appropriate Redbook series *WebSphere Everyplace Access Volume I-IV*, available at:


**HTML browsers**

The problems of supporting various browser vendors and various versions of a browser seemed to have decreased. One of the reasons might be that the pace of the vendors releasing new browser versions lessened as well. Still, the issue exists that by using certain features, your portal application will not be usable by some browsers.

“The original HTML documents recommended “be generous in what you accept,” and it has bedeviled us ever since because each browser accepts a different super-set of the specifications. It is the specification that should be generous, not their interpretation.”

Doug McIlroy

You can use tools, such as HTML TIDY (for an introduction, see [http://www.w3.org/People/Raggett/tidy/](http://www.w3.org/People/Raggett/tidy/)), to check the compliance level of your portals output in reference to the latest standards and specifications.

You are in a good position if you know what percentage of your users will use which browser.

From our experience in the field, we have the impression that more technical users switched to browsers based on the Gecko engine (such as Firefox, Mozilla, Netscape V6 and later, and others), while most others continue to use Microsoft® Internet Explorer. Due to problems of the past and to keep administration and support costs low, we also see companies with policy defining which browser the employees have to use. In general, however, by supporting and cleanly testing both Firefox and Internet Explorer Version 6, you should have already covered 80-99% of the users. Usually a minority of users have other engines, such as Opera and KHTML. If you support all four engines, you should be fine.

We do not like to see iFrames on portal pages. Years ago, the main reason was that Netscape 4.x browsers did not support it, but we do not see these browsers frequently any more. More concerns are regarding security and the portal concept. iFrames are just browsers on their own. With an iFrame in your page, WebSphere Portal would not be in control of the iFrames content, which is converse to the concept and might lead to unexpected problems. If you run

---

Note: We believe that portals with a poorly designed user interface and applications that do not leverage the mobile devices capabilities are the main factor of why there are not a lot of successful pervasive portal systems in production.

Being able to address the expectations users who want to work with their specific devices will be a key factor for a pervasive portal project.

For more information, consult the following articles about how to use WebSphere Everyplace Mobile Portal:


For more information about WebSphere Everyplace Access, consult the appropriate Redbook series *WebSphere Everyplace Access Volume I-IV*, available at:


**HTML browsers**

The problems of supporting various browser vendors and various versions of a browser seemed to have decreased. One of the reasons might be that the pace of the vendors releasing new browser versions lessened as well. Still, the issue exists that by using certain features, your portal application will not be usable by some browsers.

“The original HTML documents recommended “be generous in what you accept,” and it has bedeviled us ever since because each browser accepts a different super-set of the specifications. It is the specification that should be generous, not their interpretation.”

Doug McIlroy

You can use tools, such as HTML TIDY (for an introduction, see [http://www.w3.org/People/Raggett/tidy/](http://www.w3.org/People/Raggett/tidy/)), to check the compliance level of your portals output in reference to the latest standards and specifications.

You are in a good position if you know what percentage of your users will use which browser.

From our experience in the field, we have the impression that more technical users switched to browsers based on the Gecko engine (such as Firefox, Mozilla, Netscape V6 and later, and others), while most others continue to use Microsoft® Internet Explorer. Due to problems of the past and to keep administration and support costs low, we also see companies with policy defining which browser the employees have to use. In general, however, by supporting and cleanly testing both Firefox and Internet Explorer Version 6, you should have already covered 80-99% of the users. Usually a minority of users have other engines, such as Opera and KHTML. If you support all four engines, you should be fine.

We do not like to see iFrames on portal pages. Years ago, the main reason was that Netscape 4.x browsers did not support it, but we do not see these browsers frequently any more. More concerns are regarding security and the portal concept. iFrames are just browsers on their own. With an iFrame in your page, WebSphere Portal would not be in control of the iFrames content, which is converse to the concept and might lead to unexpected problems. If you run
into a situation where you have to use iFrames or where you strongly believe that such technologies are of advantage for your specific project, make sure that you understood the implied risks and all the necessary steps to get your iFrame working smoothly with your portal. This includes, for example, the proper session refreshing. You can learn more about that topic in the dedicated white paper written by Richard Gornitsky and John Boezeman, *The use of iFrames and Web Clipping in WebSphere Portal*, which should be available in April at the WebSphere Portal developerWorks pages:


Additionally, it is a good approach if you keep your JavaScript to minimum. JavaScript gets especially problematic if leads it you put logic in the JavaScript. Your HTML and all its embedded elements should represent the View component. The business logic should reside completely at the server side.

Cross-Site-Scripting restrictions in modern browsers prevent JavaScript inside one iFrame from accessing variables and traversing the DOM within the parent window (or other iFrames) unless they are within the same domain and use the same protocol, for example, HTTP versus HTTPS. This causes many iFrame-based applications to fail (if they rely on JavaScript, and many do). When there are Cross-Site-Scripting issues, variables end up blank, causing the JavaScript to fail or stop executing. Debugging is quite hard.

For these reasons, many existing iFrame applications simply will not work without rewriting the JavaScript.

### 3.3 Integration

Integration with other systems is the most complex and time-consuming part of a portal project. At that start of your project, build a single node of portal and integrate with the fewest systems necessary to make it functional. You will be tasked with configuration and integration with a Web server, a directory (LDAP) server, and a database server.

Configure your first system, perform load tests, and get a performance baseline. Then, build another node, cluster the two, and retest to get a second performance baseline. See 4.1, “Test” on page 80 for more about test concepts.

We have seen many installation failures due to insufficient privileges of the portal ID in the back-end system. It is most common in Oracle database environments.

**Important:** Work closely with your database administrator (DBA). Insist that the portal ID will be granted *all* permissions during setup.

Integration with back-end applications is costly because these projects tend to be complex and drag on longer than expected. Scope out these portlet development projects carefully, for example, by using the portlet sourcing method described in Appendix C, “Portlet sourcing” on page 105. You must understand the authorization model that is already in place for the back-end system before you develop a portlet. What we have observed is that every integration point tends to add another layer of bureaucracy.

Additionally, if you are new to WebSphere Portal, try to avoid introducing too many new technologies at one time.
3.3.1 Directory (LDAP) management

WebSphere Portal and WebSphere Application Server require some form of user registry. There are several possible ways to provide WebSphere Application Server and WebSphere Portal with access to a user registry:

- Lightweight Directory Access Protocol (LDAP) user registry
- Database user registry (for Portal/Member Manager)
- Custom user registry

What we have typically seen is customers using an LDAP user registry to store user information and to authenticate users. This section discusses the issues to consider if you plan to use an LDAP user registry with WebSphere Portal.

Because your LDAP will typically host either customer or employee data or both, it is one of the most important and critical components in your organization. When building a portal system, it is therefore important to include a person on your infrastructure team that is already an LDAP expert and knows very well how to extend and leverage LDAP schemas. This is not necessarily a full-time person on the team, but a person that should be readily available to you at critical points in the project. For example, this person will tweak portal configuration files to work with your LDAP schema when you are enabling security for the portal. This expert should be comfortable using an LDAP browser tool and should know your LDAP server replication strategy as well.

You will need to understand if you will install a new LDAP server or use an existing LDAP server (more common). We have seen several customers try to integrate WebSphere Portal with an unsupported version of LDAP. Verify that your version of LDAP is supported by WebSphere Portal by checking the Information Center:


You can install the LDAP on the same machine as WebSphere Portal or on a remote machine. Installing the LDAP server on a remote machine can improve performance and is therefore recommended. Be sure to have a discussion with the security architect in your organization to determine if you need to secure the data flowing between the LDAP server, WebSphere Portal, and WebSphere Application Server. If the answer is yes, you will need to set up LDAP over a Secure Sockets Layer (SSL).

A minimum of one group and one user is required for WebSphere Portal. The required group is wpsadmins or an equivalent. Members of this group have administrative authority within WebSphere Portal. It is expected that a WebSphere Portal administrative user will be a member of the wpsadmins group in LDAP, for example, wpsadmin, although Portal does not actually enforce this. If content management functions are configured, we recommend that you also create the groups wpsContentAdministrators, wpsDocReviewer, and wcmadmins.

Tip: You can use the same user ID for more than one purpose.

You will need to configure WebSphere Application Server to access the LDAP server through the Member Manager. Member Manager is the common user repository management instance for WebSphere Portal. In this configuration, one or more user registries and, therefore, one or more realm, can be created. A realm is a concept that denotes a specific body of users accessing a specific portal configuration.

Tip: The recommended configuration is for LDAP with realm support, which will allow you to create virtual portals in the future.
3.3.2 Collaboration components

Collaboration features help people in your organization work together and share information online to achieve their business goals. A collaborative portal can improve your organization's responsiveness, innovation, competencies, and efficiency. According to a survey done by META Group, 70% of those implementing a portal desire collaboration within their portal environment.

Collaborative features within WebSphere Portal include Lotus Collaboration Center, the collaborative portlets, and the Lotus Collaborative Services API. To use these features in the portal, you must set up one of more of the supported versions of the following products: Lotus Domino, Lotus Sametime®, and Lotus QuickPlace®.

The Collaboration Center is a set of pages deployed during the installation of WebSphere Portal, providing multiple customized instances of eight portlets including Lotus Web Conferencing and People Finder. These portlets depend heavily on the correct implementation of the LDAP server with the WebSphere Portal server. LDAP integration can be very tricky, especially if you have a customized LDAP schema. The Collaboration Center appears on the Workplace page of the portal.

Setting up a collaborative portal should be treated as a separate and distinct subproject of your overall portal project. It requires additional planning and configuration of not only the portlets but of the back-end servers that support them (Domino, Sametime, and QuickPlace). You must include a full-time person on your infrastructure team that is dedicated to the installation, configuration, and ongoing maintenance of the collaboration servers. Ideally, this person will have expert administration skills in Domino and be already familiar with Sametime and QuickPlace as well. If you do not have this skill on staff, you must allow time in your project plan for a person to develop these skills or hire a person for your project that does (preferred).

**Note:** Collaboration portlets require additional configuration for compatibility with external authorization products such as IBM Tivoli Access Manager and Netegrity SiteMinder.

Refer to the following Technote to understand what is involved for the integration with Tivoli Access Manager:


How WebSphere Portal uses Domino and planning your user directory are two important topics to discuss early in the planning phase of collaboration integration. Configuration varies depending on which components you are using. For example, if you plan on using portlets for both Sametime and QuickPlace, the user directory has to be an LDAP directory and they both must share that directory. This is not true if you are only using Sametime and Domino portlets. Refer to the following topic in the Information Center for more information:


You need to consider the performance and availability of Domino servers when configuring WebSphere in a Domino environment. For example, to use a Domino LDAP server as the user repository for the portal, install WebSphere Portal on a separate machine from the Domino LDAP server. A Domino LDAP server for the portal should reside on a machine that is dedicated to serving the portal environment. Note that for i5/OS®, we recommend that a specific Domino server be created to run the collaboration components and that the Domino server remains on the same i5/OS server as WebSphere Portal.
Single sign-on between the Domino environment and the portal environment enables you to log in to the portal and then use collaborative portlets without having to authenticate a second time. A best practice is to install and configure all servers prior to enabling single sign-on. Note that if you complete the required single sign-on configuration between the two environments, there is no procedure to disallow automatic logins for a specific user. For example, if user A logs in to the portal, user A will always be logged in to Domino.

If there is a non-Domino LDAP directory server in place, for example, IBM Directory Server, you can employ several strategies to integrate the existing directory with Domino and thus achieve single sign-on and awareness across any Lotus collaborative portlets. Refer to the following article from IBM developerWorks:


We have repeatedly seen projects fail at the point where the collaboration components are integrated. Do not underestimate the time and expertise needed to get this complex environment implemented correctly. You need expertise in two distinct areas: Domino and WebSphere Portal. Do not make the mistake of many customers before you and assume that one person or group can hone these skills during the project.

### 3.3.3 Traditional systems

IBM has built strong relationships with numerous independent software vendors (ISVs). Many of them have built portlets, even suites of portlets, to integrate with WebSphere Portal. Before you begin a development project integrating with an older software system, visit the IBM Workplace Solutions Catalog and check to see if the vendor has submitted portlets. The catalog is available at:

http://catalog.lotus.com/wps/portal/workplace

You will find more than 2,000 portlets here. There are many for the most popular traditional back-end systems such as SAP, PeopleSoft, and Hyperion. When you want to download a portlet, you might be directed to the vendor’s Web site. In that case, the vendor is responsible for any warranties, support, and licensing terms that relate to the portlets. You might be able to download a portlet for your project and use it right from the catalog.

### 3.4 Performance analysis

You must consider performance requirements in the planning phase of your portal system. By choosing the right topology (see also 2.6.1, “Topology planning” on page 29) and defining reasonable non-functional requirements in the service level agreements (see also 2.5, “Defining non-functional requirements as part of service level agreements” on page 23), you are part the way there.

We often see, however, that performance analysis begins after the application design was implemented or, even worse, after the first stress tests revealed poor results. Most of the performance problems we have seen are the result of poorly-designed applications.

**Tip:** Tune your portal during stress/load testing, but design your application to perform well before you start developing.

### 3.4.1 Caching

Unfortunately, it is out of scope of this best practices paper to describe in detail how you can use caching to increase performance on your portal. Here, we can only encourage you to use
various caching techniques and give our opinion about how much you can gain using which technique. Understand, however, that we are advising you from experiences based on the projects on which we have worked. The only real answer is that it depends.

There are already a number of excellent papers that describe in detail how to leverage caching for a portal system. Here are just a few:

- **Develop high performance Web sites with both static and dynamic content using WebSphere Portal 5.1**, available at:
  

- **Caching data in JSR 168 portlets with WebSphere Portal 5.1**, available at:


- **Enhancing portal page rendering performance by managing long-running back-end calls**, which includes information about the use of the dynamic cache Distributed Map, available at:


- **Creating and deploying a portlet service for IBM and JSR 168 portlets**, a tutorial that includes information about command caches, available at:


- **Using the command cache to improve portal application performance**, available at:


- **Static and dynamic caching in WebSphere Application Server V5**, a description not specifically trimmed to WebSphere Portal but with some general information about caching that is also true for portal systems:


In addition to these papers, be sure to read the related topics about WebSphere Portal and WebSphere Application Server in the Information Centers.

**Why cache?**

With caching, you can gain a lot of performance. With reasonable caching, you can:

- Prevent a request from coming to WebSphere Portal at all. This can be true for referenced parts within the portal page, such as CSS files or images, for HTML fragments, or for complete portal pages if anonymous users access the portal.

- Increase the render time of your portal page, which might lead to a substantial increase of the users’ impression of the response time.

- Avoid unnecessary rendering of portlets, which does not only speed up the response time but also leads to a reduced load of the request for the portal server. This, again, leads to better overall performance, reduced hardware costs, or both.

- Prevent unnecessary back-end calls, which again leads to a reduced load on those back-end systems and sometimes also on the portal server itself. Additionally, the time saved for the back-end calls leads directly to a reduced response time for the user.

As discussed in “Page response time” on page 27, it is very important for the overall performance of your system to reduce response time. In short, the quicker your response time, the less load is generated on the server, which provides a quicker response time. The faster responses and increased throughput increases application responsiveness, which allows for a better user experience.
You might argue the often repeated truth: You cannot tune yourself out of a bad application design. In reality, the use of caching does not just belong in the category of tuning; it should be an in-depth part of your application design process.

Unfortunately, caching comes with a downside. Its usage is also often a source of additional difficulties and complexity. In fact, the use of some caching options have been rejected in the field because a number of known defects have been detected. IBM has done a good job in correcting these, so if you have had these in previous projects, they are very likely gone in current versions.

The most important points to consider are the lifetime of the cache and the possible security breaches with data in the cache. Read more about this in “What to cache?” on page 76.

At this point, you might be wondering if you should opt out of deploying caching and instead buy more hardware. If your project is already under great time constraints, it might be the easiest option to ensure that enough hardware is available. However, if you are still in the architectural or development phase, you can consider any possibility for caching. Caching can improve your portal system by factors.

Most experienced portal consultants will say that caching is an essential piece to a portal implementation, but do not automatically assume that it is the right answer for your situation. Do your calculations. In certain rare cases, it might turn out that buying a new processor license is cheaper than paying for another work-month to implement an additional caching layer later. Therefore, the likelihood for such a situation is especially high if you choose to add caching to your architecture after you create your applications.

Where to cache?
The general guideline for caching is to hold the cached data as close to the user as possible. This starts at the Web browser. Here, we describe in detail the marked locations in Figure 3-1 and point out any concerns.

![Figure 3-1 Various locations of where to cache in your system](image)

The following numbers refer to Figure 3-1:

1. Web browser caching

   Web browser caches are a great thing because they reduce the amount of traffic to your system by magnitudes. There is a major difference if a user requests a certain static file from your portal at every request or just once a session. There is nothing special to add here from a WebSphere Portal best practices perspective.
However, although this should be well known and common best practices, we do not always see this applied. Typically, portals seem to suffer from overloaded pages. Due to the nature of a portal, on one page, there might be many portlets and many different applications underneath supporting them. Each portlet/application comes with its own user interface definition files, such as CSS and JavaScript. This can lead to portal pages with a 3-digit KB size or an unreasonable number of referenced CSS and JavaScript files. This can lead to a notable latency in rendering the portal screen. Robust client-side machines and networks might mask this problem in your development phase. Be certain to test your applications on a machine that a typical end user would have. More importantly, design your applications with these best practices in mind:

- Clean out the generated HTML output from static style sheet elements and JavaScript and move those in static .css or .js files that can be cached by the Web browser. Every byte counts.
- Remove heavy HTML comments in your HTML output. Use JSP comments instead of HTML comments in your JSPs.
- Check the referenced .css and .js files. Do they include the same style sheet, just differently named, or the same JavaScript algorithm? If so, try to agree on a single version. Note that this might not be possible for all portlets, for example, if you are using a portlet supplied by a vendor.
- Reference the smallest number of files as possible. Keep in mind that every referenced file in an HTML page needs to open a port and make a request to the server to either get the file or get the answer that the file did not change and can be used from cache. Less files lead to a better responsiveness. However, it does not make sense to have big JavaScript functions in a portlet that only a minority use in a global JavaScript file.

Remember to consider security when planning what to cache; this often gets overlooked. When developing applications, keep in mind that the user might access them from an Internet cafe or other kiosk situation. Consider the possibility of unauthorized use of a browser cache.

2. WebSphere Edge Server caching

At this entry point to your server architecture, you want to cache as many items as any possible. You do not have to use WebSphere Edge Server; it can be any reverse caching proxy. We have not seen any major issues with Edge Server working with WebSphere Portal, so this seems to be a good choice. On portal projects that do not cover all the server-side software, we see often the hosting company choose a reverse caching proxy product based less on features and more on what their administrators are familiar. Make sure it complies with your needs.

The question becomes how much is a reasonable amount to cache at this layer. The discussion centers around two main topics:

- Security

  Because your Edge Server or any other reverse caching proxy that you use is usually located in the DMZ, you do not want to keep any personalized or security-relevant data in the cache. The cache at a proxy is really a big data pool of content, and there are no security borders that prevent unwanted access to cached data on this layer.

- Invalidations

  The topic of invalidations is easily understood, but can be somewhat tricky. If you update or remove any resource in the back end, you want the change populated immediately to all of its preliminary caches. Otherwise, you might experience some odd effects.
To go any deeper here is out of the scope for this paper. However, we recommend that best practice of always choosing the secure path, which means to perform the invalidation more times than not. Additionally, make sure that you have test scenarios in place that cover any special cases that might be involved.

For more information, refer to the appropriate consultants or the WebSphere Edge Server documentation, available at:


3. Web server caching

In the past, it was common to move any static data from WebSphere Application Server to the Web server. This had and still has the advantage that the file serving servlet of WebSphere Application Server does not need to deliver the GIFs and similar files, but the HTTP server can perform this on its own. Web servers are quicker in doing this, require less resources in terms of computing power, and require at least one hop less (and sometimes more due to firewalls and other servers) from a network perspective.

Moving static files to the Web server is generally a good task (we do not want to discourage you from doing this). However, it is often not worth the time, because this process tends to lead to a lot of time-consuming organizational coordination. We have seen projects where there was a considerable amount of time invested to get every single graphic file out to the Web server. In trying to avoid a bottleneck where WebSphere Application Server needed to serve up a number of smaller graphic files, even graphics for portlets were placed on the Web server. This method can be very time-intensive because each time you need to update the WebSphere Portal wps.ear file or any portlet that comes with static data, you need this extra step. It is usually much easier and leads to less coordination if portlet developers are allowed to reference their items relative to their portlets and include it there.

The WebSphere Application Server Web server plug-in does not just do load balancing; it also does a great job of caching. Refer to the Information Center of you WebSphere Application Server (for example, http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp) to read more about how to set this up. Depending on the size of your static files, this is often a better choice than going the traditional route of copying the static files to the Web server machine. In addition to the static files, the plug-in, Edge Server, and other caching proxy products support Edge Side Include (ESI). This feature enables you to also hold HTML fragments in your cache. Depending on your portal project, it might worthwhile to explore this topic. We usually see the bigger projects leveraging this feature.

A double caching for static elements, where you cache them in a reverse proxy in your DMZ and in your Web server plug-in, does not do any harm as long as your invalidations work correctly. You might also leverage ESI only at the plug-in level in order not to worry too much about security issues that might come up by holding too much data in the DMZ (in case your Web server is located behind the DMZ).

4. HTML fragment caching

WebSphere Portal, by default, offers you the ability to configure various caching options. To understand these options, including cache expiration, be sure to study the Information Center, available at:


The internal mechanisms used for the default HTML fragment caching also rely on the WebSphere Application Server dynamic caching.
5. Leveraging dynamic caching

Because caching is considered to be defined on a J2EE level instead of on a specific API level, the topic was not considered during the definition of JSR 168, the Java Portlet API. Unfortunately, there is no publicly defined caching API by the Java community. However, you can leverage the great and partly unique possibilities of the WebSphere Application Server caching API “dynacache.” Figure 3-2 shows a brief overview of the features that come with this API to help provide you some insight as to how the parts are interconnected.

![Figure 3-2 Overview of the features and connectors that come with the dynacache API](image)

Leveraging dynacache helps reduce bottlenecks in your system. For example, assume that you detect that your traditional systems are too slow to cope with the given non-functional requirements of the system. Assume that your target response time is three seconds. You observe that a call to the traditional system by itself takes two seconds. Then, you are likely to miss your target. If, however, the data in this system is needed by many portlets and retrieved by many users, it is advantageous to cache it. The latency toward the traditional system might then only appear once every 1000 requests.

Dynacache is a feature available with the latest versions of WebSphere Application Server and WebSphere Portal. There are already a couple of excellent articles available that describe, by example, how to leverage those features. Refer to the list provided at the beginning of this section or search for even more, for example, at the developerWorks WebSphere Portal zone, available at:

The following list describes some of the issues we commonly see, including some frequently asked questions for which we have not yet found any clear answers:

- **How much data should I allow to be cached?**

  The answer is, again, it depends. Generally, the answer is as much as any possible; but how much is possible depends. One factor that you need to consider is the size of memory you have available for cache. It goes into the heap of the JVM. If your garbage collector already has trouble getting enough free memory for WebSphere Portal, you might need to consider other ways, because your garbage collector can easily use the energy you save by caching.

  Other obvious factors are the cache replacement factors, which you can recognize out of the cache hit and cache miss ratio. Here you should be able to see whether you try to cache too many items within a certain memory.

  **Note:** Where it is easy to see the number of entries in the cache monitor, it seems to be tricky to obtain the size of the individual objects.

- **Is it better to include many small objects or fewer bigger objects?**

  Try to leave out the bigger objects. They consume a relatively large amount of space in memory, so the ratio of saved requests per available MB of cache is far better for smaller objects.

  In addition, from a user's perspective, you might not mind waiting for a large PDF 10 or 20 seconds longer, instead of waiting for each single request to WebSphere Portal just a single second longer.

- **Is it better to leverage cache disk-offload or do less caching?**

  This is a question of how expensive (in terms of money and time) your back-end calls are and of how much memory you have still available in heap to be used for cache data. As an option, you might consider increasing the heap size.

  Just be aware that reading from a fast database can be quicker than reading from a very large disk cache. It will be up to you to find the right cache size for your applications.

- **Do I want to use cache synchronization between the nodes in a cluster setup?**

  To synchronize cluster data over various nodes in a WebSphere cluster is a very powerful feature and thus it is important to use it carefully. We see two main issues with it:

  - The amount of data replicated between the node members can be an issue. Usually not because of the load that is put on the network (because current networks are very fast), but more because of the load on the WebSphere Portal JVMs. This is especially noticeable during the startup of a JVM, when it begins to synchronize with all other members. It is important to note that this process pushes a substantial load on the already running JVMs. Be careful when in a production scenario.

  - Depending on your setup, you might find that you gain or lose by having cache data local to each node member. The amount of members you have in your cluster environment is not as significant as how much memory you have available for caching and how expensive your back-end calls are. This is expensive not only in the increase in time but also in money, because you might have the situation where each call to a certain system costs x amount. Replication might save, at a theoretical maximum, the factor of node members you have. If replication leads, however, to a higher cache replacement, you are on the downside again.
– When would I want to use a cache domain?

A cache domain can be of help if you run require cache replication but also discover that point-to-point replication might lead to a large performance drain. Whether to go with one or the other option depends on your setup and your applications. And again, check carefully if you do not have a fast performing database in place. Buying in the complexity of a cache domain should pay off in performance improvement.

**Note:** Do not forget that you can configure your caches individually in order to have some synchronized and some not.

– Can I just use the session to cache some user data instead of using dynacache?

In some rare cases where you really need to cache some data scoped to the user, you might need to hold the cache keys in the session to allow referencing. Some of the reasons include:

- By concept, the session data should only be altered in the action/event phase of a portlet. To do otherwise is not just bad design, but might also lead to problems.
- You do not have the configuration option to replicate the data within a cluster, independent of your decision regarding session sharing.
- The size of the session grows unnecessarily.
- You mix data of different values. As cached data is replicated, session data is not.

**Note:** Do not misuse the session for caching.

– Because dynacache is easy to use, can I use it to share data between my portlets?

No. Dynacache was designed and created to cache data. Although it is easy to leverage, this is not a reason to misuse its concept and implementation.

In summary, depending on your portlet applications and portal system, leveraging dynacache is a technique where you have the chance to really gain a large amount of performance. In addition, some portal sites are not able to live without it.

**Note:** WebSphere Portal also leverages dynacache for its internal caching strategies. For a cluster scenario, it replicates some cache data between the cluster nodes.

6. Database/back-end caches

This is out of scope for WebSphere Portal and therefore also for this best practices paper. Consult your database administrators and database specialists. Being able to tune your back ends in one way or another, however, often proves to be a major factor in the overall performance of the system. Therefore, it is worth checking whether there is the possibility to either add a cache or add, for example, more indexes to your database.

**Tip:** Test the performance of your database back end even if you are not in charge of setting up and configuring the database back end.
What to cache?
Similar to “Where to cache?” on page 70, the answer here has just as many possibilities. The question that remains is “What is possible?” or, because you want to cache as much as possible, “What would you not want to cache?”

As already discussed “WebSphere Edge Server caching” on page 71, there are two main problems with using caching techniques, security and invalidation:

- Security

  Personalized data can only be cached on a personalized level, for example, in the browser and in cache keys bound to a specific user session. This is not a good idea, because the browser cache the data is not really secured and manipulating cache data in the session is not recommended.

  Therefore, the best way to secure data is not to cache data that needs to be secured.

- Invalidation

  As already discussed in “WebSphere Edge Server caching” on page 71, invalidation can be a tricky topic. Again, you want to make sure with reasonable test scenarios that whatever you cache you can securely invalidate again and that the reload cycles (cache timeouts) are not too high. With this process, even in the case of a non-working invalidation, you will be back on track after a couple of minutes. However, the time in between this might appear to the user as an outage.

WebSphere Application Server comes with a cache monitor tool that you can use to view the static files and HTML fragments that are in the cache. This monitor consists of a couple of JSPs that call the appropriate WebSphere Application Server APIs. We have, therefore, also seen customers that extended these JSPs to their needs, which seems to be a good idea to us.

Tip: Unfortunately, we have seen many projects lately with performance problems that could have been avoided if the vast possibilities of caching were considered.

However, do not substitute a focus on caching strategies for functionality and clear design. A good design might still outweigh any caching ideas, or in the words of Tom Alcott, “You can’t tune or clone your way out of a bad application design.”

3.4.2 Sessions

This section provides additional information about how to treat sessions within a WebSphere Portal environment. Guidelines need to be recalculated for every specific project, but these should help you to apply best practices.

Portal and portlet sessions

There is often some confusion regarding portal and portlet sessions. In essence, the portlet session objects are namespace encoded session objects that are embedded within the portal session. There were also some changes taking place when moving to the JSR 168 portlet API. For a good overview of the differences, see:


The importance of size

One of the most frequently asked questions about the earlier WebSphere Portal portlet API was how to cast the portlet session down in order to be able to read and write to the global HTTP session. It was a reoccurring question because the answer changed over time due to
changes in the underlying WebSphere Application Server versions. A common add-on to this question was, “Why is such an essential functionality not included in the standard API?” The answer is because of its original design and because there is a good chance that it gets misused. Therefore, it was common to see it used as a shared memory to exchange data between portlets. This is not a good use because it can lead to massive data in the sessions.

There was the argument that you can save data in the session by allowing a global scope, especially if more than one portlet has the intention of saving the same data in the portlet sessions. This was ultimately addressed in the JSR 168 portlet API, where you can address data in the session now in a global or private scope. However, do not use it, for example, to send messages from one portlet to the other.

The size of a portal session will naturally be larger than the best practices numbers that a standard WebSphere Application Server-based application suggests. This is true because a WebSphere Portal server session on its own, excluding portlets that put additional data to the session, uses about 4-5 KB. This number was already reduced in the latest releases. For more information, see: http://www.ibm.com/developerworks/websphere/techjournal/0505_col_hines/0505_col_hines.html

Because portlets sometimes include the complexity that previously whole Web applications had on their own, the overall session size sums up to factors for the global session size. This is not a best practice because portlets are supposed to be far more lightweight. Session sizes that are too large will eventually lead to problems. As a rule, you might want to target a single-digit number. This is not hard number that you have to reach in order to have a successful portal project. However, no matter how your portal system is configured, we believe this is a reachable number.

Session sharing in a cluster environment
Assuming that you want to use session sharing, you can either share the session through your database or use memory-to-memory replication. For a closer description of both techniques, refer the WebSphere Application Server Information Center, available at: http://www.ibm.com/software/webservers/appserv/was/library/

You might assume from reading this description that you would experience a great deal of performance improvement by using memory-to-memory replication, because you do not need to go to your database to persist the session there. In reality, however, we do not recognize a real noticeable boost because most of the computing time that is relevant for performance goes into serialization of the session object instead of persisting it.

Years ago, on portal projects based on early WebSphere Application Server V5 products and earlier, we recommended that customers did not use the memory-to-memory replication because we observed a large number of problem reports. If a new product feature does not give you much of an improvement, always select the well-known and often-used approach. Naturally, you always suspect such a new feature in case of a problem, even if there is not a rational basis to do so. Having too many new features at once might, therefore, cost you valuable time during critical problem determination.

With Version 5.1.x and later of WebSphere Application Server, we see more and more customers adopting the technology of memory-to-memory replication. Another way of analyzing which technology to deploy is to evaluate how much control you have over your environment. For example, if you use database persistence, but do not have full control over the database, you might experience delays during development and while troubleshooting. However, a valid reason for choosing database persistence might be that you run into trouble with the heap size of your JVMs. Every session adds to the heap of every single portal JVM. Large sessions and a large number of concurrent sessions might influence your decision.
Chapter 4. Deploying, testing, and maintaining a portal

In the final chapter of this best practices paper, we focus on the topics that have been the reason for many critical situations in various projects. While in previous versions of WebSphere Portal (up to Version 4.x) most of the problems concerned deploying the software in a clustered environment, this has changed starting with version 5. The product became much more mature regarding installation and deployment, and the procedures were also more clearly documented by the improved WebSphere Portal Information Center and various excellent articles. Today, we believe the number one reason for project delays is improper test procedures. In addition, we have the impression that the period of time after going live with a portal is often not covered sufficiently. We discuss these items so that hopefully you do not encounter these issues.

For more information, see:

4.1 Test

As developers know, a considerable number of their working hours are spent testing, possibly more than developing on its own. Modern project management guidelines state that poor testing (for example http://www.coleyconsulting.co.uk/failure.htm) is one of the five key reasons why projects fail. Skyler Thomas, an IBM Senior Technical Staff Member, states “The number one reason for deployment delays and failures is an improper test strategy and environment.”

You must regard testing as one of the most important items in your portal project.

The following sections provide the guidelines on which you should focus.

4.1.1 Test processes and environments

A common question we hear is when should testing start. The answer is when you start to create anything on your project. This also includes the installation of environments. We discuss non-functional tests in 4.1.2, “Non-functional tests” on page 83.

Unit tests

When developers start to create any code on their workstation, they must test on a unit test layer. To have this included as a must do in the development process is already a first step for testing.

Mandatory unit testing is a cornerstone of a good development project. Developers should test early and test often. Everybody will agree with this statement; however, the cost of a well-developed testing framework is high. JUnit helps with testing. JUnit is a concept, a framework of Java classes, and a run time, developed by Extreme Programming. Mandating JUnit tests for all back-end calls can be very helpful. Refer to the following link for more information regarding Extreme Programming:

http://www.extremeprogramming.org

Unit tests are also a recommended process within Rational Unified Process® (RUP®)-driven development. Refer to the Rational Web site for more information:


Unit tests and especially the JUnit framework are well supported by Rational developer tools. Understand that the recommended system requirements for these tools need to be taken seriously. Therefore, your developers will require well-equipped workstation hardware. We generally recommend these tools because we have good experience with them for portal developments. However, if you cannot afford, for example, the 2 GB minimum for each workstation, do not use these tools.

Daily and weekly builds for integration tests

As soon as the developers are satisfied with a certain code file, they should mark it accordingly in the central version control system. We see it as absolutely mandatory that your developers share their code in a central version control system. We often see that customers leverage the Concurrent Versions System (CVS) (http://www.nongnu.org/cvs/), which is released under the GNU General Public License and runs on almost all operating systems. Rational Software has with the various versions of IBM Rational ClearCase also a version control systems in its portfolio. For more information about Rational ClearCase, see:

The code maintainer would now create scripts that compile the sources that are in the version control system and create deployable packages. Each day, the code maintainer would create a daily build and deploy it to the integration development environment, and weekly, the maintainer would do the same with more mature, marked code to the tester integration environment.

We recommend establishing the role of a code maintainer. This person would probably not need to be at the project the first day that the developers start, but understand that it will take some time to have scripts ready that automate the process. Depending on the size of the project, it often turns out to be a full time job.

In some cases, it might be required to have an additional environment that enables the developers to directly deploy some of their components. This is, for example, true if the developers have an insufficient test environment or no test environment at all on their local workstations. We have seen time savings by using VMware (http://www.vmware.com) images for such environments. If developers deploy code on their own, they might not be sure about how to revert cleanly. Having the policy of creating snapshots before using those environments will ease backups and prevent the time-consuming activity of finding problems.

Figure 4-1 illustrates how the developers add their code to the central version control system, from which the code maintainer creates the daily builds that go the developer integration environment and the weekly builds that go to the test integration environment. You do not want to combine the developers’ and the testers’ integration environments, because these have different targets.

![Figure 4-1 Overview of the test environments](image)

Developers require such an integration environment to see their code working for the first time, integrated with the code of the other developers in the project. Here, you will often recognize things such as class loader problems (sometimes less experienced portlet
developers might assume that they can put libraries in central locations, such as the WebSphere Application Server /lib directory). This helps highlight dependencies, which developers might not think about when deploying code on their local workstation. Therefore, a dedicated team member must oversee the deployment. In such an environment, developers might also see their code working with real back-end test environments for the first time. On their local workstation, they would typically write the code against some stubs or dummy test environments, because it is often not possible to grant every developer direct access to a back-end test environment.

Testers will expect a more mature and less often changing environment for their function tests (versus daily deployments). Because cycling through some test cases might require quite some time, it is not an ideal situation if the code basis changes every 24 hours, possibly leaving the system unavailable for two hours a day (due to deployment and possible deployment problems). One week a frequently chosen time frame. Depending on your project, you might find an even more reasonable time frame for your project.

Every problem record the function testers detect will be reported in a bug tracking tool. They will mark the build version they used while reporting the problem. Make sure that the developers know to which code version in the version control system to which the build belongs. In addition, establish methods that ensure that the right developers get assigned to the reported bugs and that they are solved by priorities. There are a number of commercial bug tracking tools available, and lately, we have seen the frequent and successful use of the open source tool Bugzilla, available at:

http://www.bugzilla.org

Management and business process tests

Do not underestimate the importance of keeping your top management and your project sponsor happy. In Figure 4-1 on page 81, we also added a layer for business tests that you rarely find in general postoperative documents.

Because portal projects are usually integration projects that tend to be big and thus cost-intensive projects, your management might be especially concerned about the progress of the project. In addition to delivering regular progress reports, we learned that the acceptance is higher if they are able see a pilot or test system as soon as any possible. Do not worry about disappointed reactions to the current functionality, because they will understand that user interaction will be slow and functionality incomplete, but they will honor the progress they can actually see.

If an additional demonstration environment does not cost too much time and money, it is often worth to establish it, even if it is just for the sake of your top management. Be careful promising demonstrations without such an environment. You can "lend" any of the environments to do some demonstrations, but understand that your team will lose quite some time by freeing that environment and ensuring that the demonstration works. Having a demonstration environment enables people to experiment with a certain set of almost stable components at any time.

In addition to a demonstration environment, we recognize that there is sometimes a requirement for a business process test environment. This can be similar to the demonstration environment, only for a very small, single server portal environment. We have seen companies that were required to ensure that a certain business process that had a different user interface still works correctly using the portal. Other than the functional tests, these tests are more dedicated toward overall scenarios. While the function tests ensure that the right output is returned based on a certain input, these tests are concerned with the systems involved in the background, for example, to ensure that the requests to certain traditional systems do not lead to corrupt or inconsistent data over time, or assume a request
triggers a workflow that leads to a printed letter for the customer. In such cases, the function tests might not evaluate all systems involved in the workflow. Therefore, we see if long-running (in terms of days or weeks) transactions are involved. You do not want to hold any environment for such a long time, but these testers do not want to have anything changed during that time. Therefore, it is good to provide a special environment, even if it is not fully equipped.

**Note:** Every additional environment requires additional time for your administrators. Every person on your project waiting for an environment to leverage causes possible delays in your project.

### 4.1.2 Non-functional tests

As a natural prerequisite for non-functional tests, you will require the proper hardware and environments. We discuss what you should do to get started and then continue by elaborating about what you can do if the results are not as expected.

**Prepare for non-functional tests**

While we see that functional tests are relatively well prepared at most portal projects, projects suffer and sometimes even run into long delays because of insufficient non-functional tests. Is it difficult to write well-performing portal components? Not very.

One reason might be that the procedures of functional tests do equal usual J2EE projects. As you can see in 4.1.1, “Test processes and environments” on page 80, we do not mention portal specifics often. Portal projects do, however, usually involve complex environments, because these projects target the integration of environments. This again leads to a high level of complexity, which adds difficulty for any part of an IT project (including security, as described in “Security concepts” on page 37). We can try to reduce complexity (no user, no back-end systems, no cluster, and so on), but this is converse to portal projects goals. After carefully reading Chapter 2, “Planning a portal” on page 13, you might have already recognized that we try to push in that direction to save you money and reduce project risks.

Now with an architecture in place with a certain topology and a certain application, you have to make sure that this is all going to work as you defined in your service level agreements (see 2.5, “Defining non-functional requirements as part of service level agreements” on page 23).

**Environments**

One of the most frequently asked questions about non-functional tests is if it is possible to use the integration environment for some non-functional tests. As you can see from Figure 4-1 on page 81, these environments have distinct goals. Different people are working on the environments, and the environments are also of different hardware sizes.

The load test environment must be a mirror of the production environment. Of course, it might have fewer CPUs. In some cases, it might also have less memory. It might even have less machines within a cluster.

**Important:** It is important that the load test environment is a mirror of the production environment and must, therefore, match the production environment topology.

You will not get representative results if your load test environment is not equal in topology. To save costs, projects often put all components on a big single box and do their load tests on
that environment. We know that budgets are tight everywhere, but doing your load tests in such a way simply wastes money.

If you build a completely new system, it is sometimes possible to leverage the environment that will be later used as a production environment for your load tests. This provides a big advantage in that you can compare your results of your load tests with the real results just after you went into production. You will never be able to simulate the real world in your load tests with 100% accuracy, but it is good if you are able to create test scenarios that leverage your components with only a 20% discrepancy. Some items, such as user behavior and connection line delays, will always be hard to reproduce. If you perform your tests on an absolutely equal environment or the production environment itself, you will have good numbers to compare. Unfortunately, this does not prevent you from requiring a load test environment.

Important: You need a load test environment.

A portal project will always be an iterative project, and the iterations might happen earlier than you expect them. In addition, you do not have the chance to save time and build up a load test environment after you have gone into production. Then, you lose the advantage of comparing figures with load tests on an environment that is equal to your production environment in size.

Stress tests
Because some customers worry about how the portal system will perform, they understand why load tests are required. Stress tests are done, where the word “stress” is implemented literally. While this type of a task force gets high management attention, the tests themselves are sometimes designed poorly. For example, we have seen a test generator that was quickly programmed by one of the developers. It hit the portal system with an enormous amount of requests, neither analyzing the responses nor caring about possible user behaviors.

Building reasonable load tests requires experience. Here, we mention a couple of important parameters:

- User behavior such as “think time.”
- Login process of users and the percentage that explicitly log out or get implicitly logged out by session invalidation.
- Application usage, for example, which applications are most frequently used.
- Transaction usage, such as how many transactions get submitted to the back end (for example, a user might use an application often, but frequently cancel before submitting).

In addition to these general factors that non-functional testers would know, we add a best practice that we learned in our portal projects. In many cases, it was helpful to do a so-called baseline test and then add as few components as any possible.

This means that you should first do the load test right after the installation of the base software components. This enables you to have a test available that you can compare directly to the tests that we do in our software laboratories. If these numbers differ substantially from the numbers that you got from Techline sizing, this may indicate a problem. Are your tests correct? Are there any problems in the environments, for example, on a network or operating system level? Do not continue unless you have some convincing answers.

The following example describes this clearly. We got called to a customer because of unresolvable performance problems for his portal. While analyzing the system, we detected that even a static HTML page on the HTTP server was suffering with the same performance problems as the whole portal. In the end, it became clear that a wrong router configuration
was the reason that the system was not able to keep up with the requested response time. These are unlucky situations that lead to massive time delays.

**Note:** Testing to failure means that you *assume* that things will break and that you will need to fix things.

Baseline tests are often not done because non-functional tests start too late in a project phase and creating them is considered to be a throw-away asset. The test scenario on a default WebSphere Portal installation is different than the test scenarios on your custom portal systems and there is not a lot of reuse possible. We believe it is worth the time.

Continuing from there, you should apply an iterative approach here as well. Add as few components as possible. For example:

1. Start with a non-clustered environment and an out-of-the-box WebSphere Portal implementation.
2. Cluster your portal system.
3. Exchange the portals default security system (LDAP with a special schema) with your security system, but do not any portlets yet.
4. Exchange the portals themes with your custom themes, but do not add any of your portlets yet.
5. First add those portlets that seem to be critical to you from a performance point of view.

Do not continue until you are convinced that you know what is happening and why the numbers changed the way they did. Make sure that you document your results in a way that anybody (for example, a performance specialist) will understand it several weeks later.

**Importance of load tests**

At this point, you might worry about what proper performance tests require and wonder if it is really necessary to do these test. Unfortunately, they are, however, necessary.

At this point, you might already balance, if it is really worth to go this stony path. Unfortunately, there is no way to come around it.

Stress testing is not a negotiable item. Your portal *will* fail in production if you do not do adequate stress testing before the production release. Remember, eventually your portal will touch all your back-end IT systems. Many of these systems will have increased utilization rates because the portal will, at the very least, increase awareness of applications, if not expose the user to added functionality.

**Testing for the service level agreements**

If you followed our advice in 2.5, “Defining non-functional requirements as part of service level agreements” on page 23, you defined acceptable performance requirements in your service level agreements. You used Techline sizing to estimate the proper hardware configuration. Now, you are ready to deploy your first application and you need to perform a stress test making sure that it *exceeds* the performance requirements. Load test the environment to the point of failure. Tweak individual components to achieve optimum performance, including every portlet.

**Tip:** It is good to outperform the negotiated service level agreements, but because tuning requires time and money, consider it as an extra.
Load test tools
There are a number of tools available that support your stress tests. Some of them are expensive, but provide excellent features. One load test tool is Mercury LoadRunner. For more information, see:


Rational also provides load test tools, for example, the Rational Test Suite. For more information, see:


Resolving performance problems by code reviews, profiling, and more
Usually, at the point where a project detects that it will not meet the required performance, code reviews and profiling is requested. We discuss this topic by starting with a couple of lines of code shown in Example 4-1.

Example 4-1  A small performance issue

```java
class MyList extends List {
    public void add(Object o) {
        this.add(o);
        this.trimToSize();
    }
    public void remove(Object o) {
        this.remove(o);
        this.trimToSize();
    }
}
```

This might be a bit too easy, but it was found at a large B2C portal project, where experienced developers were working, and it had a definite impact on the garbage collector.

What happened? At every request, a large number of objects where added to a big list of the type `MyList`. Because the list was always just right in size, each additional object required the Java implementation to allocate new memory, copy all objects over to the newly allocated memory, and lose the previously used memory. The Java implementation is smart enough to allocate a bit more memory in order to make sure that this does not happen the next time an object is added. However, the `trimToSize()` made sure that additional bit is just cut off again.

It is very likely that the developer thought that the list is going to be big and set the `trimToSize()` to save memory. What this developer did not know or understand when writing the code was that the method gets called multiple times during a single request. Just because of this method, each request was allocating MBs of heap to just throw it away immediately afterward. Because it was a high-volume portal that had to support more than 100 concurrent requests per second, the portal JVMs were straining under the load. No tuning helps remove a small line of code in the wrong place.

Developers also make mistakes, but unfortunately we do not think that it was the developer's fault. To be clear here, it is not a good idea to put `trimToSize()` within the add/remove method, but we see a different trend. We see many projects that do not seem to be very well organized. Skills, especially development skills, are requested like a shopping list. Developers come in, get quickly set up, write their code, and are pushed to another project. Therefore, components are developed by different people, having different perceptions of the code's usage. In addition, sometimes parts of the development goes to off-shore developers, which
might lead to a setup through telephone calls and requirement papers that were quickly
cobbled together.

We are not proposing that off-shore development is necessarily a detriment to your projects.
Instead, we want to describe the risks that you buy into with such an approach. The likeliness
that your portal does not perform as assumed by the Techline sizing are higher as well as the
risk for project delays. If you do not insist on clean working code, you might pragmatically
calculate your costs and risks.

For example, every developer and IT architect will usually tell you that it is not necessary to
buy new hardware because Portal does not match your performance requirements. However,
a pure calculation might unveil that it is sometimes a valid option. This might be true if you are
running out of your projected development time and you intend to discard your code, extend
your code for the next iteration, or if you intend to add more hardware. Therefore, you gain the
chance to first add the hardware and then solve the problems in the code.

**Important:** To summarize this, investing in a good development culture pays off at the end
of a project.

At this point, it should be clear that extensive tests are very valuable. However, let us
continue with the assumption that you have now such a situation and discuss how best to
proceed.

A code review of the project might help, but is usually overestimated. At the upper example,
an experienced reviewer would have detected that the garbage collector allocates lots of
memory per request and would have then, for example, by profiling, looked for the classes
responsible for allocating memory during a request. This can require an extensive period of
time, the proper tools and environments, and perhaps a bit of luck. Therefore, if you do not
have the proper environment now, you are losing even more time and money.

Often code reviews happen on a more general basis. Due to the usually large amount of
code, the reviewer will only be able to check certain subsets of the code. The code will be
compared to general best practices and some hints and tips are given. We have rarely seen
that this has shown major effects on projects. We rather tend to see it as an instrument for
project management to underline their opinion that it is not the code and to show that they
have managed their project well.

So does profiling help? If it is done well, it can help. The advantage of profiling is that it can be
effective if it is done by people who do not know what the portal system is supposed to do.

The correct approach is to do profiling as early as possible. The developers on their own
should profile their code. In reality, often they do not get the time to do it, because time lines
are generally tight. It requires some experience to discover what parts of the code should get
a high priority regarding profiling and which parts are of less interest. This gets more difficult,
because with WebSphere Portal, you have a product that brings much functionality with it.
Therefore, people get confused by the number of classes and request flows. This is, however,
the wrong approach. While profiling, try to exclude all classes that belong to the WebSphere
Portal product. You will not find any problems in these classes. In addition, many profiling
tools break if you try to profile a whole WebSphere Portal implementation.

A very easy way to enable profiling is to use the IBM Java built-in tracing facility. It leverages
the Java Virtual Machine Profiler Interface (JVMPI) hooks and requires less processor and
memory resources than any other “regular” profiling tool. For a detailed description of the
trace features and functionality, refer to the *Java Diagnostics Guide*, available at:

**Tuning**

Before you start tuning your portal environment, refer to the *IBM WebSphere Portal Version 5.1 Tuning Guide* for more information, available at:

http://www.ibm.com/support/docview.wss?uid=swg27005771&aid=1

In general, we recommend that you only tune parameters when you understand their impact and thus are fully aware of what you are doing.

Furthermore, within a portal system, there are many more components other than WebSphere Portal and your applications that might be a possible bottleneck, for example, the network or the operating system. The components around authentication, such an LDAP server, are often even more important. Database tuning is another key element to remember.

4.1.3 After going live

If the project does not end the day the portal system goes live, tests need to continue as well.

**Staging and preproduction environment**

At this point, we also discuss the staging and preproduction environments that we included in Figure 4-1 on page 81. These are combined in some projects, and you will need them by preparing next iterations before you can move new items to the production environment. On these environments, you also test the deployment of the new items. Therefore, never add anything manually, because you would not add anything manually to the production environment. This is your last step to ensure that your deployment scripts work correctly.

An example where you would not want to combine the preproduction with the staging environment is if you have a business process that require some preproduction stage. The staging environment should be dedicated to updates that include some technical change, such as adding another portlet.

You need your load test environment and you must not combine it with the staging environment. The environments have different targets and are generally used by different people. Saving costs here can seriously hurt the maturity of your portal system.

One way to use a preproduction or production-test environment is to use it for enhanced demonstrations, for example, to preview new possible features. In this case, you might want to leverage a "portal jail" using Web Services Remote Portlet to provide these features. A portal jail would not be a trusted portlet server for portlets that follow a less stringent QA process. Using Web Services Remote Portlet, you then still comply with security and quality insurance guidelines and you have a perfect environment for producing demonstrations to customers or lines of business within your organization. It can also be a very effective way to try out the "what if" scenarios or to try out new portlets in a production-like scenario.

You might argue that it is cost intensive to require these environments the whole time and that you naturally need some consolidation. In cases where everything works perfectly, this might be a true statement, but in reality, we have the experience that this does not work out successfully. A distinct load test and staging environment is required.

**Resources**

An interesting thing we detect is that after a portal system goes live, the people involved with the project leave the project. The work is done, but it also continues. Developer resources and test resources are replaced by maintenance resources.
Chapter 4. Deploying, testing, and maintaining a portal

It is a rare project where maintenance team members or the developers of the next iteration do not complain about insufficient documentation. It will never be possible to have enough documentation. There are so many things happening during a project that even the best documentation might not cover, for example, which components were often the source of problems or what back-end systems appeared to deliver strange responses even though these responses were never resolved.

Therefore, in a best case scenario, always keep some original people in the project. With some leaving for new challenges and some coming in with new ideas, the quality is kept high. Although it might be difficult, we also believe that it helps if people are not declared as exchangeable resources that can be removed and added like any other on demand hardware component.

4.2 Release planning

Use the 80/20 rule when planning a portal project. Figure 4-2 illustrates this rule. Focus on delivering 80% of the project's functionality by connecting to existing systems usingportlet builders, such as WebSphere Portal Application Integrator, and by avoiding more difficult integrations, such as using the Web Clipping portlet. Rebuild only 20% of the most useful functionality from the existing Web site.

---

“I’ve never met a human being who would want to read 17,000 pages of documentation, and if there was, I’d kill him to get him out of the gene pool.”

Joseph Costello, Recipient of the Phil Kaufman Award 2004

---

Figure 4-2   Use the 80/20 rule for your project
4.3 Deployment

Here is an example of how you can implement a portal build process across your environments:

1. A developer implements portlets, servlets, Enterprise JavaBeans™, and other J2EE artifacts using either WebSphere Studio or source code that is delivered into a version control system.

2. A designer creates themes, skins, HTML pages, portlet JSPs, and other design elements using any editor.

3. The results are delivered into a version control system.

4. An administrator creates the content tree (labels, URLs, and pages) using the WebSphere Portal administrative user interface of a development portal.

5. The resulting content trees and portlet instances are exported using XMLAccess or a script and then delivered into a version control system.

6. The release manager assembles a consistent release in the version control system and creates the delivery. The release manager executes scripts (for example, ANT) to extract Java sources, design elements, and configurations from the version control system and then runs a build (compile and package).

7. The operator takes delivery and deploys it onto the staging and production systems. The operator executes ready-made configuration tasks (for example, ANT, XMLAccess configurations, and wsadmin scripts) to deploy the delivery.

The chart in Figure 4-3 depicts a typical build process across multiple environments.

![Portal Deployment Diagram]

*Figure 4-3  Portal deployment across environments*
**Determining what to move**

To deploy a portal release, resources must be synchronized, packaged, and deployed to a portal server. This is a manual process that involves many people and few tools. Resources can be:

- **Portal configuration** (this is stored in the portal database):
  - Portal content tree
  - Portal application configuration, settings, and data
  - Portlet access control information
- **Portal artifacts** (these are stored in the file system):
  - Portal configuration (property files)
  - Theme and skin file assets (JSPs, style sheets, images)
  - Portlet code (Java classes, JSPs, XML files)

**Important:** You must define and test your application deployment process.

### 4.3.1 Automating custom code deployment

Although there is no automated method to move portal applications from one environment to another, there are two options:

- **Completely replace the old release with a new release.** The drawbacks of this option is that any data that was customized by the user is lost. Although this option works, we do not recommend it.
- **Use the XMLAccess tool to load incremental or differential releases.**

### 4.3.2 Staging concepts

A subsequent solution release is staged from the integration to the staging and to the production system. The physical implementation of configuration staging through a series of systems does not really move these configurations between systems. This process is based on repeatable modifications of portal solution releases on multiple systems. For each solution release, a differential portal solution configuration is imported into the system; artifacts are managed manually by manual updates and deletions.

What elements of the portal you move depends on the type of release you have. If you have an incremental release, you:

- Add new resources to a release.
- Update resource attributes (only add properties to lists).

If you have a differential release, you:

- Maintain all functionality of the incremental release.
- Delete existing resources.
- Update resource attributes (add or delete properties in lists).

If you have data that has been configured by the user, configure the scope of the portal for a single user.

The following process (Figure 4-4 on page 92) is an example of a possible subsequent portal solution staging process. Derivations of this process are possible and expected. It focuses on configuration and artifact management.
4.3.3 Clustering and deployment

In WebSphere Application Server, a cluster is composed of multiple identical copies of an application server. A cluster member is a single application server in the cluster. WebSphere Portal is installed as an enterprise application server within the WebSphere Application Server infrastructure. All of the clustering features available within the WebSphere Application Server infrastructure are also available and apply to WebSphere Portal. Therefore, a WebSphere Portal cluster is simply a collection of multiple WebSphere Portal servers that are identically configured. In Figure 4-5 on page 93, we can see a typical portal environment with clustering.

We see many improvements with WebSphere Portal clustering in the latest version. For example, WebSphere Portal configuration tasks are cell aware. This means that tasks can determine whether the node is federated and then act appropriately. Nodes running on different operating systems are supported in the same cluster. The activate-portlets task can be used to activate portlets across all cluster members at one time.

Setting up clustering is one of those features that can be difficult, even with these improvements and better documentation. If you are new to WebSphere Application Server and have never clustered an application in the past, we suggest that you enlist the help of IBM Services. A good working knowledge of WebSphere Application Server is needed to achieve success with this endeavor.

Important: Do not release your portal code from the staging or the production environment before sanitizing all logs. Deploy a zero tolerance for warnings and errors and stay with it.
As we saw in 2.6.1, “Topology planning” on page 29, there are a wide range of configurations for WebSphere Portal. At this point, you know that WebSphere Portal is a complex product that spans much of your infrastructure. Therefore, making WebSphere Portal highly available is a small project all by itself. In this section we recommend point you to additional documentation and describe some of the most common unexpected difficulties.

In WebSphere Portal V5.1, there are two ways to create a cluster of WebSphere Portal servers. There is an easy way and a more complex way. The first way (easier) is to create a WebSphere cell with multiple federated nodes first and then install WebSphere Portal onto each node in the existing cell. The second way is to start with a set of stand-alone nodes with WebSphere Portal already installed on them and then federate them into a cell. What do we recommend? Well, that depends.

There is a caveat to the easier method of using an existing cell of application servers and then installing WebSphere Portal into that environment. When you federate an application server node into a cell, it loses its default configuration and inherits the configuration held by the Network Deployment Manager. For example, if there is an issue with the cell and you have to un-federate the node, the original configuration is loaded from a backup copy and the node reverts to its previous configuration. So, in this example, when you un-federate it, you have a default application server and the WebSphere Portal application is lost. You need to think carefully about this option, even if it is easier to configure up front.

With the more complex method, where you install WebSphere Portal on each node and then federate it, the WebSphere Portal application becomes part of the node’s default configuration and cannot be “lost” even if it gets un-federated. As we said, this way is more...
complicated and we have seen users make many mistakes. Before you begin, read the Information Center:


Then, use this excellent guide, *A step-by-step guide to configuring a WebSphere Portal V5.1 cluster using WebSphere Application Server V5.1.1.1*, available at:


We have seen customers struggle with some of the following issues.

When attempting to deploy portlets in a clustered environment, you get error messages such as “Cannot install the selected WAR file.” To enable portlet deployment in the cluster, you must edit the DeploymentService.properties file on each node and set the wps.appserver.name property to the name of the cluster. See *A step-by-step guide to configuring a WebSphere Portal V5.1 cluster using WebSphere Application Server V5.1.1.1* for more information.

Each node in the cluster should have the same synchronization settings to ensure consistency between the WebSphere Portal server configurations on each node. Refer to the guide to check this setting.

After federating a WebSphere Portal node and then attempting to access the portal, you receive an Error 503 “Failed to load target servlet [portal] message.” To correct this, update the deployment manager configuration for the new portal node. Also, be sure the CellName property in the wpconfig.properties file is set for the new portal. Refer to this link for more information:


You need to enable dynamic caching on the cluster member nodes to correctly validate the portal caches. If you do not, users might have different views or different access rights, depending on which cluster node handles the user’s request. Refer to the guide for more information.

**Important:** You should create a replicator entry on each node to provide replicator failover. This prevents the scenario where the node with the replicator goes down, preventing other nodes from being able to start because they cannot access the replicator.

The more replicators you have defined in your environment, the better the replication fail-over capability will be. In the event that an application server process on which the replicator is defined is unavailable or goes down, there will be other replicators available to fill the gap. However, because additional replicators will impact the overall performance of your environment, carefully plan the total number of replicators needed.

For best performance, you can also provide a completely separate system running a dedicated application server instance as the replicator host. This dedicated application server instance need not have WebSphere Portal installed on it, although it must be in the same cell and in the same replication domain as the WebSphere Portal cluster. For more information about using replicators, refer to the WebSphere Application Server Information Center:

For more information, also see the excellent IBM Redbook, *IBM WebSphere V5.1 Performance, Scalability, and High Availability WebSphere Handbook Series*, SG24-6198:

http://www.redbooks.ibm.com/abstracts/sg246198.html

**IBM Workplace Web Content Management users**

If you intend to use the Web Content Management function included with WebSphere Portal in a cluster environment, additional configuration is required. Refer to the “Cluster Installation Process” topic in the *IBM Workplace Web Content Management Version 2.5 - Installation Guide* for information about using Web Content Management in a cluster (http://www.lotus.com/ldd/doc/ufiles.nsf/docs/WCM25/$File/WebContentManagement-2-5-InstallationGuide.pdf). In addition, pay specific attention to information for the following areas:

- Data repository. Unlike WebSphere Portal nodes that share the same database in a clustered environment, Web Content Management installations require a separate data repository, even when used in a cluster.
- Authoring portlet. Although the Authoring portlet cannot be used to create content in a cluster, you must still install the Authoring portlet on all cluster nodes to support syndication and caching.
- Secondary node. When adding a secondary node to a cluster, you must run the update-wcm-cluster-configuration task on the node.
- User registry. If you change the user registry for Web Content Management (for example, by configuring for an LDAP directory), you must run the update-wcm-wmm task.
- Web server. If you are using an external Web server with your cluster, you must run the modify-wcm-host task on each node in the cluster.

To support search in a clustered environment, you must install and configure search for the remote search service on a WebSphere Application Server node that is not part of the WebSphere Portal cluster. For more information about using search in a cluster, refer to 2.6.7, “Search” on page 50.

**4.3.4 Keeping track of the growth**

Your organization will likely have multiple e-business problems to solve, thus will most likely have multiple portals to host and support. As your expertise matures, so will your portals. Because WebSphere Portal integrates with existing applications, such as customer relationship management (CRM), enterprise resource planning (ERP), and sales force automation (SFA), we have seen customers struggle with a rapidly growing, complex infrastructure. Portals have a tendency to grow over time, potentially doubling or even tripling in function and size. A number of best practices for maintaining complex systems exist, and the software architect must take full advantage of these in order to manage this type of complexity and growth. One is creating a portal solution based on a methodology for leveraging large reusable assets. Refer to the following IBM Redbook *Architecting Portal Solutions*, SG24-7011, for best practices about design patterns and architecture:

http://www.redbooks.ibm.com/abstracts/SG247011.html

Another best practice is to apply maintenance on your servers on a timely basis. How will you know if there are updates to apply? It is easy with your IBM support ID and IBM self-help portal. IBM uses WebSphere Portal for the site:

http://www.ibm.com/support

In 1.7.1, “Register on the IBM software support Web site for Passport Advantage” on page 8, we explain how to register for this site. Be sure to choose the link My support and
personalize your preferences. Here you can add which IBM products for which you want e-mail notification. Each week, you will receive an e-mail that lists new technotes, releases, and fix packs. Figure 4-6 shows this site.

![My support](image)

**Figure 4-6  My support**

### 4.4 Maintenance

You will need to perform minor maintenance on your WebSphere Portal installation at some point. By minor, we mean small software fixes, such as database fix packs, WebSphere Application Server interim fixes, and software point releases, while maintaining 24x7 availability of the portal. You can do this, but a bit of planning is needed. Refer to the Information Center for step-by-step procedures:

```
```

Assumptions for maintaining 24x7 operation during an upgrade process include:

- WebSphere Application Server distributed session support is enabled to recover user session information when a cluster node is stopped for maintenance.
- Load balancing must be enabled in the clustered environment, and multiple HTTP servers must be available to provide Web server fail-over support.
- The portal cluster has at least two horizontal cluster members.

The situation will also arise when you need to perform maintenance to the components that make up your clustered portal environment. A component can be a Web server, a database...
server, or even a directory server. You will want to service these without bringing your portal
down. This is possible, and you can learn how by following the advice in this article:


**Monitoring**

Your ability to respond quickly to portal performance problems is key to ensuring and
maintaining a successful portal project environment. We have seen real-time monitoring of
WebSphere Portal become one of the biggest concerns in the field. System administrators
are pressured to monitor and manage performance and availability of J2EE and WebSphere
applications efficiently. Monitoring tools will help operations staff, administrators, and
developers resolve bottlenecks quickly. You need to think about monitoring more than just the
portal. Back-end systems such as content management servers and databases need to be
monitored. Focus on database capacity, throughput, and response time. Front-end systems
such as HTTP servers and caching proxies cannot be ignored. The key is to quickly
determine and isolate the problem. For example, determine if the issue is within the portal,
one of your custom applications, or within the supporting infrastructure.

We strongly encourage you to evaluate, select, and deploy a tool. There are many products
and vendors from which to choose. Each one claims to be the best solution. Tools can be
broken down into three major categories:

- **Development and profiling/performance solutions**
  - JProbe, IBM Rational PurifyPlus™

- **Application server and J2EE monitoring solutions**
  - IBM WebSphere Studio Application Monitor, Wily, IBM Tivoli Performance Viewer, IBM
    Tivoli Composite Application Monitor

- **Monitoring management frameworks**
  - Candle® Omegamon suite, Mercury Performance Center

This section focuses on the application server and J2EE monitoring solutions. There are vast
differences among the tools offered here. Determine the type of tool you are after. For
example, do you just need a tool that simply analyzes Web site usage? Are you already a
WebSphere Portal Extend customer? If so, IBM Tivoli Web Site Analyzer might be a good
place to start. Or are you looking for a tool that can integrate into your existing Tivoli
Monitoring or Rational tools? Then, one or perhaps both of the IBM Tivoli Composite
Application Monitor products is the answer. Here, we describe some of the tools we frequently
see deployed, with the exception of the Tivoli Composite Application Monitor products, which
are new in the marketplace. If you purchased WebSphere Portal Extend, you are entitled to
install the IBM Tivoli Web Site Analyzer product. All other tools mentioned are available at an
additional cost.

Monitoring tools need access to a wide range of data to provide usable results. There are
several ways of accessing data from WebSphere applications:

- **Performance monitoring infrastructure**
- **JVMPI**
- **Byte code instrumentation**
- **Application request metrics**

All the tools discussed in this section use one of these approaches, and some use more than
one. Most tools use an agent to manage client/server topology to collect, correlate, and
display data.
Performance Monitoring Infrastructure (PMI) is a client/server-based production level monitoring solution. The server collects PMI data in memory, for example, the servlet response time or data connection pool usage. The data points are then retrieved using a Web client, a Java client, or a Java Management Extensions (JMX™) client. Most of the following tools use this interface for a least part of the data they collect. WebSphere Application Server includes the PMI client Tivoli Performance Viewer.

Important: Depending on what PMI features are turned on, the performance impact of PMI itself can be slight or substantial.

The Java Virtual Machine Profiler Interface (JVMPI) is a JVM-level interface that enables the collection of data about the JVM itself. For example, it can collect data about garbage collection, JVM memory usage, thread information, and object allocation. JVMPI is a two-way function call interface between the JVM API and an in-process profiler agent. The JVM API notifies the profiler agent of various events, such as heap allocations and thread starts. The profiler agent can activate or deactivate specific event notifications, based on the needs of the profiler. The Tivoli Performance Viewer leverages a JVMPI to enable more comprehensive performance analysis.

Important: Java Virtual Machine Profiler Interface moderately increases the performance impact.

Byte code instrumentation can be broken down into two distinct areas: application-level instrumentation and server-level instrumentation. Instrumentation of application classes can be performed at run time as classes are loaded. Data is also collected at the method level. This technique is used by Wily and Tivoli Composite Application Monitor. Server level instrumentation can be performed on specific WebSphere Application Server classes by adding a monitoring hook. This can be done dynamically or by rebuilding specific classes. This technique is used by Wily, Tivoli Composite Application Monitor, and WebSphere Studio Application Monitor.

IBM Tivoli Web Site Analyzer
Tivoli Web Site Analyzer is a Web application that captures and analyzes Web site data to provide useful reports about visitor traffic, visitor behavior, site usage, site content, and site structure. Support for WebSphere Portal includes specific report elements that enable you to analyze portal usage data, such as ranking of the portal pages viewed by visitors and portal login trends. Refer here for more information:


Important: IBM Tivoli Web Site Analyzer was withdrawn from the market on July 13, 2005. However, you can still obtain support for this product until IBM withdraws support for WebSphere Portal Extend V5.x.

IBM Tivoli Performance Viewer
Tivoli Performance Viewer is a Java client that retrieves the Performance Monitoring Infrastructure (PMI) data from an application server and displays it in a variety of formats. We see most customers use this tool for a quick look. You can view data in real time with it and in chart form, allowing for a visual comparison of multiple counters.

Tivoli Performance Viewer provides advice for tuning the system for optimal performance and gives recommendations about inefficient settings. With Tivoli Performance Viewer, you can report on enterprise beans, EJB™ methods, servlets, Web container pool, Object Request
Broker (ORB) thread pool, and the connection pool. Refer to the Information Center for more information:


**IBM Tivoli Composite Application Monitor for WebSphere**

IBM Tivoli Composite Application Monitor for WebSphere is the follow-on product to WebSphere Studio Application Monitor that launched on November 8, 2005. It provides enhanced monitoring for WebSphere Portal. It is more robust and consumable than WebSphere Studio Application Monitor and can optionally interface with other Tivoli products. It was designed for application support, test, and development teams to help you gain deep insight into the health of your production and preproduction environments by using key performance metrics that pinpoint the source of bottlenecks or other defects in application code. Improvements include:

- Better coverage of portal pages and portlets, including reports for both portlets and portal pages.
- Additional and improved portal nested request types and contextual navigation. The six nested request types are Page Loading, Page Rendering, Model Building, Portal Topology, Authentication, and Authorization.
- Advanced historical reporting of key portal performance trends with deep-dive tracing for root cause problem determination.

Tivoli Composite Application Monitor for WebSphere runs on these platforms: IBM AIX 5L V5.2 and V5.3, Sun™ Solaris™ 8 and 9, Microsoft Windows® 2000, 2000 AS, and 2003, Red Hat Enterprise Linux 3.0 (IBM @server pSeries, xSeries, iSeries™, and zSeries®), Red Hat Enterprise Linux 4.0 (pSeries, xSeries, iSeries, and zSeries), SUSE Linux Enterprise Server 8 (pSeries, xSeries, iSeries, and zSeries), SUSE Linux Enterprise Server 9 (pSeries, xSeries, iSeries, and zSeries), and HP-UX 11i v1.

Tivoli Composite Application Monitor for WebSphere supports the following databases: IBM DB2 UDB V8.1 and V8.2, and Oracle 9i V2 and 10g.

Learn more about this new product here:


---

**IBM Tivoli Composite Application Monitor for Response Time Tracking**

Tivoli Composite Application Monitor for Response Time Tracking is the follow-on product for Tivoli Monitoring for Transaction Performance. It can proactively recognize, isolate, and resolve transaction performance problems using robotic and real-time techniques. It is an end-to-end transaction management solution that monitors end-user response time and helps you visualize the transaction’s path through your application systems, including response time contributions of each step.

Tivoli Composite Application Monitor for Response Time Tracking helps you to adopt an end-user’s perspective to monitor and measure performance. It follows the application’s path to help speed problem resolution. It automatically learns the environment and establishes response time thresholds. It will help you to validate the service level delivered to the end user.

For more information, refer to this link:

**Wily Introscope**

Wily Introscope enables you to monitor complex Web applications end-to-end. Use it to manage mission-critical applications from the browser to application components to back-end systems. Introscope can ensure manageability of the entire portal workflow in production. It can also isolate problems with individual portlets. Introscope provides you with the capability to create custom dashboard views of your entire application infrastructure, including Java applications, application servers, Web servers, messaging middleware, databases, and transaction servers. Refer here for more information:


**SurfAid**

SurfAid™ is a services offering that has three options. The Executive Metrics product offers Web site analytics and reporting. The Publishers product adds the COUNTER code of practice onto the metrics. An advanced tool called Analysis adds ad-hoc queries and the ability to create dynamic reports onto the Metrics product. All SurfAid products work by transferring Web log files to a SurfAid facility on a daily basis. The data is mined and the results are stored in a relational data warehouse. Reports are generated and data is queried using a Web interface. Refer to this link for more information:


**Ascera Manager 5 for Portal**

Ascera Manager for Portal is a production monitoring and diagnostic tool. It has a Discovery engine that automatically discovers new WebSphere Portal servers in a cluster. It can optionally integrate with other network management tools such as HP OpenView, Tivoli, CA Unicenter, and BMC Patrol. Ascera's goal is to monitor complex portal applications from the top down, focusing on business units of work and subprocesses.

For more information, refer to the following Web page:

http://www.acsera.com/section/view/wces/88/

**Summary**

While all of the tools provide good real-time analysis of what is happening, they do not provide any insight into “what is normal” (except for Tivoli Performance Viewer wizards.) There is no easy answer to this question because it can be very application and environment dependent. Good performance evaluation generally requires the help of an experienced WebSphere Application Server or WebSphere Portal person, or both. We have not looked deeply at specific performance measures because of the breadth of this subject. Again, this requires deep experience in J2EE, WebSphere, and WebSphere Portal to assist with this effort. The performance degradation on most of the tools discussed here appears to be 3-5%. Deeper monitoring (or sampling more often) can affect these estimates. We have seen empirically that running production systems at very, very high load and using a monitoring tool can alter system performance and perhaps even cause system crashes.
Sample workshop agenda

Table A-1 provides a sample agenda that can be modified to suit a particular customer's needs and environment.

<table>
<thead>
<tr>
<th>Date and time</th>
<th>Topic</th>
<th>Topic leader</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day 1: Requirements and portal capabilities</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8:45 – 9:00</td>
<td>Introductions and agenda</td>
<td>Customer/IBM</td>
</tr>
<tr>
<td>9:00 – 10:00</td>
<td>Business requirements</td>
<td>Customer</td>
</tr>
<tr>
<td></td>
<td>Portal project team structure</td>
<td></td>
</tr>
<tr>
<td>10:00 – 11:00</td>
<td>Portal application requirements</td>
<td>Customer</td>
</tr>
<tr>
<td>11:00 – 12:30</td>
<td>Existing system architecture and integration approaches</td>
<td>Customer</td>
</tr>
<tr>
<td>Lunch</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1:30 – 2:30</td>
<td>Technical requirements (performance, availability, and so on)</td>
<td>IBM</td>
</tr>
<tr>
<td>2:30 – 4:00</td>
<td>Portal features summary (demonstration optional)</td>
<td>IBM</td>
</tr>
<tr>
<td>Day 2: Architecture</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9:00 – 12:00</td>
<td>Portal architecture best practices</td>
<td>IBM</td>
</tr>
<tr>
<td></td>
<td>Logical architecture</td>
<td></td>
</tr>
<tr>
<td>Lunch</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1:00 – 4:00</td>
<td>Architecture white boarding session</td>
<td>IBM</td>
</tr>
<tr>
<td>Day 3: High-level application design</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9:00 – 11:00</td>
<td>Portal application design best practices</td>
<td>IBM</td>
</tr>
<tr>
<td></td>
<td>Development structure, roles, responsibilities</td>
<td></td>
</tr>
<tr>
<td>Date and time</td>
<td>Topic</td>
<td>Topic leader</td>
</tr>
<tr>
<td>---------------</td>
<td>----------------------------------------------------------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>11:00 – 12:00</td>
<td>Portal design/interaction, specific portlets (list)</td>
<td>IBM</td>
</tr>
<tr>
<td></td>
<td>Lotus collaborative portlets</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Content management</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Other &quot;canned&quot; portlets</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Custom portlets</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Lunch</strong></td>
<td></td>
</tr>
<tr>
<td>1:00 – 4:00</td>
<td>Portal application white boarding</td>
<td>IBM</td>
</tr>
<tr>
<td></td>
<td><strong>Day 4: Portal development and follow-up discussions and generate documentation</strong></td>
<td></td>
</tr>
<tr>
<td>9:00 – 12:00</td>
<td>Project plan review and risk assessment</td>
<td>IBM</td>
</tr>
<tr>
<td></td>
<td><strong>Lunch</strong></td>
<td></td>
</tr>
<tr>
<td>1:00 – 4:00</td>
<td>Portal operations and deployment considerations</td>
<td>IBM</td>
</tr>
<tr>
<td></td>
<td>(administration, monitoring, portal solution release process)</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Day 5: Follow-up discussions and generate documentation</strong></td>
<td></td>
</tr>
<tr>
<td>9:00 – 12:00</td>
<td>Workshop wrap-up</td>
<td>IBM</td>
</tr>
</tbody>
</table>
Sample portal tracking worksheet

In this appendix, we discuss a sample portal tracking plan that you can use for your portal planning. Due to the size of the tracking worksheet, we provide a link to the source.

This worksheet assists in tracking the progress of the project without using a full-fledged project plan or software. This highly customizable document helps you list and estimate pieces of work very early in the project. Then, use this data to assign work to different developers on the team and to track their progress during the project. This approach provides the following benefits:

- A spreadsheet is very easy to use and maintain, unlike the more formal project plan which quickly goes stale and is never easily updated.
- Project managers appreciate when you have initial tasks and estimates early in the project for them to feed into their project plan.
- The form can be used to feed into design and to track the progress of the team. The lead developer or architect is usually the one to build and maintain the spreadsheet. This lead can use it to assist in breaking out different components (portlets, services, and other components) that will be needed in the design.
- The form is compact (usually one or two pages) that can be carried around and updated with the current status as you talk to team members. Once or twice a week, the changes can be incorporated into the document and a new one printed. This will probably occur daily in the early stages of the project, that is, in the design phase.

Refer to the following Web page for this worksheet:
Portlet sourcing

The appendix includes a portlet sourcing exercise and worksheet.
Exercise

Consider adding our portlet sourcing process to yours. It can save you a lot of time (two months on average).

Portlet sourcing is a critical activity for a successful project.

The benefits of portlet sourcing include:

- The primary basis for accurate project sizing
- A great way to communicate your organizations needs effectively
- The primary source of portlet requirements for developers

This exercise describes the three step portlet sourcing process:

1. Start by identifying a default Web page.
2. Next, we identify which areas we want to include in our portal. We give each portlet a label, as shown in Figure C-1.

Figure C-1  Portlet sourcing: The portlets are identified and labeled
3. This document describes each portlet needed for this page. See the sample in Figure C-2.

Research and record items such as the data provider, state, and personalization requirements.

See Appendix B, “Sample portal tracking worksheet” on page 103 for a worksheet for use in your next project.

Allow ample time to complete the portlet sourcing exercise. Typically, plan a week for wire frames and a half day per unique portlet. Use wire frames and state transition diagrams to document and validate the user experience.

One of the most difficult items to ascertain is how to bring content into the portal. This exercise will help you to identify:

- The content source (where the content is and who owns it)
- How you are going to get the content into the portal

Be sure to include end users in your planning sessions and not just the business analyst. This might save rework time because the end users are the audience you will be trying to please in the pilot phase.
Worksheet

Table C-1 provides a portlet sourcing worksheet.

<table>
<thead>
<tr>
<th>Portlet sourcing worksheet</th>
<th>Hints</th>
<th>Your response</th>
<th>Your comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summary information?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Portlet title?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Portlet title?</td>
<td></td>
<td>News feed.</td>
<td></td>
</tr>
<tr>
<td>Portlet ID?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Portlet ID?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ownership?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>This worksheet completed by?</td>
<td></td>
<td>John Smith.</td>
<td></td>
</tr>
<tr>
<td>Portlet development assigned to?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Portlet codeveloper/reviewer (if any)?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Portlet tester?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Target portlet completion date?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Navigation?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Location (on which pages)?</td>
<td></td>
<td>Home.</td>
<td></td>
</tr>
<tr>
<td>Primary model?</td>
<td>Cooperative (affects other portlet's states), or wizard (changes state/appearance based on user interaction).</td>
<td>Wizard.</td>
<td></td>
</tr>
<tr>
<td>Sends/receives messages with which other portlets?</td>
<td>Includes Click-To-Action and portlet messaging.</td>
<td>No.</td>
<td></td>
</tr>
<tr>
<td>Access control?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Who can see it?</td>
<td>All users.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Who can modify configuration data?</td>
<td>Administrators.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Who can place it on pages?</td>
<td>Administrators.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Content?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Single or Multiple items displayed on portlet?</td>
<td>Multiple.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>If multiple, number of items to display?</td>
<td>5-10 news items.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sort order of content (or random)?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Filtering: By date?</td>
<td>No.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Filtering: By user? (See personalization below.)</td>
<td></td>
<td>No.</td>
<td></td>
</tr>
<tr>
<td>Portlet sourcing worksheet</td>
<td>Hints</td>
<td>Your response</td>
<td>Your comments</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------</td>
<td>---------------</td>
<td>---------------</td>
</tr>
<tr>
<td>Uses people awareness?</td>
<td></td>
<td>No.</td>
<td></td>
</tr>
<tr>
<td>Uses Click-To-Action?</td>
<td></td>
<td>No.</td>
<td></td>
</tr>
<tr>
<td>Content includes links?</td>
<td>To more detailed information.</td>
<td>Yes.</td>
<td></td>
</tr>
<tr>
<td>If links, expected behavior when clicked?</td>
<td>Content appears in new window.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Customization/personalization?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>At entire (concrete) portlet level?</td>
<td>That is, display or do not display portlet based on LDAP group membership.</td>
<td>Everyone sees.</td>
<td></td>
</tr>
<tr>
<td>At portlet content level?</td>
<td></td>
<td>Using user-configured selections.</td>
<td></td>
</tr>
<tr>
<td>Multiple language support?</td>
<td></td>
<td>One language.</td>
<td></td>
</tr>
<tr>
<td>Languages?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multiple device support/markup languages?</td>
<td>One markup language.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Markup languages?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data owner/source?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Owner (name/phone) of data?</td>
<td>Subscription through XXX and then select which news items pertinent to the company.</td>
<td>Web group.</td>
<td></td>
</tr>
<tr>
<td>SLA in place with owner?</td>
<td></td>
<td>No.</td>
<td></td>
</tr>
<tr>
<td>Risk of change in data format?</td>
<td>Slight.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Risk of availability problems?</td>
<td>Moderate.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Development?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Portlet source?</td>
<td></td>
<td>Configured.</td>
<td></td>
</tr>
<tr>
<td>If “Catalog,” name and navcode from catalog?</td>
<td>Catalog: Typically minimal configuration required.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>If “Configured,” type of portlet?</td>
<td>Configured: No Java development, but requires significant configuration.</td>
<td>Web Clipping.</td>
<td></td>
</tr>
<tr>
<td>If “Developed,” tooling to develop it?</td>
<td>Developed: Developed using Java tooling.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Needs to support JSR 168?</td>
<td></td>
<td>No.</td>
<td></td>
</tr>
<tr>
<td>Portlet sourcing worksheet</td>
<td>Hints</td>
<td>Your response</td>
<td>Your comments</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-------</td>
<td>---------------</td>
<td>---------------</td>
</tr>
<tr>
<td>Needs to support Web Services Remote Portlet?</td>
<td></td>
<td>Yes.</td>
<td></td>
</tr>
<tr>
<td>Number of states?</td>
<td>“States” are different appearances of the same portlet as you interact with it; you can have one JSP per state, for example.</td>
<td>No.</td>
<td></td>
</tr>
<tr>
<td>Modes/window states?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>View?</td>
<td>Yes.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Configure?</td>
<td>No.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edit?</td>
<td>No.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Help?</td>
<td>Yes.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Print?</td>
<td>Not really a mode, but do you need to be able to print just this portlet?</td>
<td>No.</td>
<td></td>
</tr>
<tr>
<td>Special (different view) for maximized state?</td>
<td>Maximized state might want to take advantage of more screen space.</td>
<td>No.</td>
<td></td>
</tr>
<tr>
<td>Special (different view) for solo mode?</td>
<td>“Solo” is the rendering of a portlet in a browser by itself, typically without header and navigation elements.</td>
<td>No.</td>
<td></td>
</tr>
<tr>
<td>Testing?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Test data source?</td>
<td>Production data.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>If test data, does that data source have to be created?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Is portlet or data source likely to be performance-sensitive?</td>
<td>Multiple queries of multiple databases of data that cannot be cached are a performance-sensitive portlet. Be sure to test the portal with and without performance-sensitive portlets to understand their impacts.</td>
<td>Yes.</td>
<td></td>
</tr>
<tr>
<td>Caching?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cacheable?</td>
<td>Yes.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Portlet sourcing worksheet</td>
<td>Hints</td>
<td>Your response</td>
<td>Your comments</td>
</tr>
<tr>
<td>------------------------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------</td>
<td>---------------</td>
<td>---------------</td>
</tr>
<tr>
<td>Same cache for all users?</td>
<td></td>
<td>Cache is same for all users.</td>
<td></td>
</tr>
<tr>
<td>Will you cache data feed?</td>
<td></td>
<td>Yes.</td>
<td></td>
</tr>
<tr>
<td>Will you allow caching in portlet.xml?</td>
<td></td>
<td>Yes.</td>
<td></td>
</tr>
<tr>
<td>Will you use the WebSphere Application Server dynamic caching system (dynacache)?</td>
<td>This enables command caching of database queries, dynamic Java object caching of JSPs, servlets and so on, and edge caching.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Solution assurance checklist

This document is to be completed by the customer and meant to assist the IBM subject matter expert (SME) and the quality assurance specialist in preparation for and during an IBM Solution Assurance Review.

In the following tables, include the names of specialists responsible for approving the item or providing information, use cross references to the action items list at the end of this document, and include brief comments if appropriate, for example, “specialist x has read the proposal and confirms that the proposed solution meets the requirement.” If there is no information available for a particular requirement, the review should assume the requirement will be onerous/high.
### Table D-1  Solution Assurance Review checklist: Customer requirements

<table>
<thead>
<tr>
<th>Activity: Customer requirements</th>
<th>Y/ N</th>
<th>Factors increasing risk</th>
<th>Factors reducing risk</th>
<th>Task owner</th>
<th>Due date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer organization and executive sponsor identified?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Assumptions and customer risks understood and documented?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Customer's present business environment/background understood and documented?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Customer's baseline requirements:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>▶ Functional</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>▶ Operational</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>▶ Performance</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>▶ Scalability and workload understood, documented, and signed off?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>▶ Business process integration requirements understood and documented?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>▶ Web content management requirements understood and documented?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conditions of satisfaction and acceptance criteria are known, documented, and agreed?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Business benefits and drivers understood and documented?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Staffing/resources available to support this solution?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Discuss a readiness plan with the customer before implementing WebSphere Portal V5.1.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table D-2  Solution Assurance Review checklist: Product capability

<table>
<thead>
<tr>
<th>Activity: Product capability</th>
<th>Y/ N</th>
<th>Factors increasing risk</th>
<th>Factors reducing risk</th>
<th>Task owner</th>
<th>Due date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Have the overall <em>architecture and software products</em> needed to support the solution been determined?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Can WebSphere Portal meet the customer's <em>availability, scalability, and workload</em> requirements? Are the requirements documented and signed off?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Can the published capabilities and functionality of WebSphere Portal server meet the customer's general <em>expectations</em>?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Review of Information Center for product capability. Contact a Techline specialist through the sales or marketing branch to size the customer environment or call Techline Phone Support at 1-888-426-5525.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table D-3  Solution Assurance Review checklist: Solution evaluation criteria

<table>
<thead>
<tr>
<th>Activity: Solution evaluation criteria</th>
<th>Y/N</th>
<th>Factors increasing risk</th>
<th>Factors reducing risk</th>
<th>Task owner</th>
<th>Due date</th>
</tr>
</thead>
<tbody>
<tr>
<td>What kind of project is being proposed?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>► First portal installation.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| ► Migration of previous version of WebSphere Portal to V5.1; refer to the migration section of the Information Center:  
  For the migration guide, go to:  
<p>| | | | | | |
|                         |     |                         |                       |            |         |
| ► Migration of another vendor’s portal to 5.1: |     |                         |                       |            |         |
|   ▶ If a migration, are plans for migration of processes in place? |     |                         |                       |            |         |
|   ▶ Have all other product-related Solution Assurance Reviews been completed? |     |                         |                       |            |         |
|   ▶ Has the customer reviewed all available migration information for migration planning assistance? |     |                         |                       |            |         |
|   ▶ Have all migration tools for WebSphere Portal been reviewed? |     |                         |                       |            |         |
|   ▶ Does the customer have a fallback plan if migration is not successful? |     |                         |                       |            |         |
| Is the customer an existing Experience customer? |     |                         |                       |            |         |
|   If so, is the customer aware that the Experience offering has been withdrawn and what they are now entitled to? Consult the local sales team if in doubt. |     |                         |                       |            |         |
| Will the portal involve: |     |                         |                       |            |         |
|   ▶ Migration of non-WebSphere Web application to portal model? |     |                         |                       |            |         |
|   ▶ Migration of WebSphere Web application to portal? |     |                         |                       |            |         |
|   ▶ New application? |     |                         |                       |            |         |
|   ▶ Migration of large amounts of data from one database/source to another and is there a plan to handle it? |     |                         |                       |            |         |</p>
<table>
<thead>
<tr>
<th>Question</th>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>What types of functionality will the portal use?</td>
<td></td>
</tr>
<tr>
<td>- Customized out-of-box portlets (excluding Web Clipping).</td>
<td></td>
</tr>
<tr>
<td>- Web Clipping.</td>
<td></td>
</tr>
<tr>
<td>- Inter-portlet communication.</td>
<td></td>
</tr>
<tr>
<td>- Portal document management.</td>
<td></td>
</tr>
<tr>
<td>- Virtual portal. <strong>(Note:</strong> If realm support is required, be aware that multi-LDAP support will not be available until a post-GA fix pack).**</td>
<td></td>
</tr>
<tr>
<td>- Business process integration.</td>
<td></td>
</tr>
<tr>
<td>- IBM Workplace Web Content Management (if the response is Y, include the SAR guide for Workplace Web Content Management V2.5 and V5.1 with this review).</td>
<td></td>
</tr>
<tr>
<td>- Web services. Support for Web Services Remote Portlet V1.0 is included in WebSphere Portal V5.1. See the Information Center for more information. For more details about Web Services Remote Portlet 1.0, refer to: <a href="http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrp">http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrp</a></td>
<td></td>
</tr>
<tr>
<td>- Web services proxy portlet.</td>
<td></td>
</tr>
<tr>
<td>- Credential vault.</td>
<td></td>
</tr>
<tr>
<td>- HOD portlet. Is the HOD server installed and configured? Is the communications server needed?</td>
<td></td>
</tr>
<tr>
<td>- HATS portlet.</td>
<td></td>
</tr>
<tr>
<td>- Prebuilt portlet from catalog access to back-end systems.</td>
<td></td>
</tr>
<tr>
<td>- Custom portlet development access to back-end systems.</td>
<td></td>
</tr>
<tr>
<td>- Lotus Extended Search.</td>
<td></td>
</tr>
<tr>
<td>- Other portlets shipped with WebSphere Portal (specify).</td>
<td></td>
</tr>
<tr>
<td>- Other portlets from the portlet catalogue (specify). Is their functionality (and any limitations) known and understood?</td>
<td></td>
</tr>
<tr>
<td>Are customer requirements for branding the look and feel of the portal known? Customizing themes and skins might be a non-trivial piece of work.</td>
<td></td>
</tr>
<tr>
<td>Will JCA adapters be required? Are they pre-written or will they need to be developed?</td>
<td></td>
</tr>
<tr>
<td>Will WebSphere MQ connectivity be required?</td>
<td></td>
</tr>
<tr>
<td>Will pervasive devices be used to connect to the portal?</td>
<td></td>
</tr>
<tr>
<td>Are unusually severe consequences possible if the solution implementation is unsuccessful or delayed, or is the solution considered to be mission critical?</td>
<td></td>
</tr>
</tbody>
</table>
Appendix D. Solution assurance checklist

Table D-4   Solution Assurance Review checklist: Design requirements

<table>
<thead>
<tr>
<th>Activity Design requirements</th>
<th>Y/N Factors increasing risk</th>
<th>Factors reducing risk</th>
<th>Task owner</th>
<th>Due date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Does the design meet the functional requirements?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Does the design meet the performance requirements?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Does the design meet the availability requirements?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Does the design meet data backup and recovery requirements?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Does the design meet upgradability requirements?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Does the design meet scalability requirements? See the clustering section of the Information Center.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Has anyone been involved with a similar project before?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Is a benchmark or proof of concept required? If so, has it been resourced and funded?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Does the design or solution depend on an unannounced product? List which products.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>▶ If yes, has the appropriate risk analysis been performed?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>▶ If yes, has the customer signed a non-disclosure agreement (NDA)?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Is the solution complex?

Has a project leader with overall responsibility for coordinating this project been assigned?

Has an overall technical leader, with overall responsibility for development of the project, been assigned?

What stage is the project currently in?
Consider the scope of the required solutions assurance appropriately:
▷ Preliminary research
▷ Design and architecture
▷ Development
▷ Test and deployment

Are you aware of or do you anticipate any conditions that might impair your ability to deliver the solution successfully?

Are you aware of any conditions that might reduce the likelihood of feeling very satisfied with the solution, even if it is successful?
Table D-5  Solution Assurance Review checklist: Configuration

<table>
<thead>
<tr>
<th>Activity: Configuration</th>
<th>Y/N</th>
<th>Factors increasing risk</th>
<th>Factors reducing risk</th>
<th>Task owner</th>
<th>Due date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Have all fixes noted in the release notes been downloaded?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Proposed IBM software and hardware configured and documented?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Proposed non-IBM hardware and software configuration documented and verified (might require a safety review and/or disclosure agreement)? <strong>Note:</strong> If Citrix software is part of the solution, and Internet Explorer running on top of a Citrix server is to be shared between Citrix clients, this is <em>not</em> a supported WebSphere Portal client. You can run Internet Explorer side-by-side with a Citrix client on the user’s workstation.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Have the configurations been produced or checked by a specialist experienced in using the configurator?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Have the appropriate versions of all required software been included?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Has software and hardware compatibility been checked?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Has the overall architecture and application flow of the solution been determined?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Have the performance, scalability, load balancing, and high availability requirements been documented?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Do you have an active network connection that supports TCP/IP? ▶ Yes, to an intranet. ▶ Yes, to the Internet.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Are there other software products, (excluding prerequisites and corequisites) that will be integrated into the solution (both IBM and non-IBM)?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Activity: Solution characteristics</td>
<td>Y/N</td>
<td>Factors increasing risk</td>
<td>Factors reducing risk</td>
<td>Task owner</td>
<td>Due date</td>
</tr>
<tr>
<td>---------------------------------------------------------------------------------------------------</td>
<td>-----</td>
<td>-------------------------</td>
<td>------------------------</td>
<td>------------</td>
<td>----------</td>
</tr>
<tr>
<td>Have the security requirements been documented, (for example, firewalls, authentication, user authorization, and so on)?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Root user, or a user with root authority must be used to install much of the software. Does the customer allow using root for installation purposes?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Does the user who will be performing the installation have permission to create/update users and databases? Need to have requisite permissions for LDAP and database.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Will a proxy server be involved in the installation?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Communication ports:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>▶ Are all communication ports that will be used, understood, defined, and documented?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>▶ Will all ports be open during software installation?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Have you built adequate time into the project schedule for load/scale/performance testing?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Is overall raw performance an issue?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Which version control system is to be used and does the customer have the skills required to administer the system?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>▶ CVS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>▶ ClearCase</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>▶ Other (specify)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>▶ No version control</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Will workflow be used in Workplace Web Content Management?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Will workflow be used in Portal Document Manager?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Is the solution or application required to be up and running 24x7?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Have you consulted the IBM Support Web site to see if there are any reported problems or available fixes/fix packs that might affect the success of the project?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IBM Support: <a href="http://www.ibm.com/support">http://www.ibm.com/support</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Activity: Solution components</td>
<td>Y/N</td>
<td>Factors increasing risk</td>
<td>Factors reducing risk</td>
<td>Task owner</td>
<td>Due date</td>
</tr>
<tr>
<td>------------------------------------------------------------------------------------------------</td>
<td>-----</td>
<td>--------------------------</td>
<td>-----------------------</td>
<td>------------</td>
<td>---------</td>
</tr>
<tr>
<td>Have you identified the Web server to be used as part of the solution?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>▶ Yes, but the specific vendor and version have not been checked against the list of supported Web servers.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>▶ Yes, and the Web server is one of the supported Web servers.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>▶ Yes, but the Web server is not one of the supported Web servers.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Have you chosen the database that will host the portal configuration data?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>▶ Yes, but the specific vendor and version have not been checked against the list of supported databases.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>▶ Yes and the database is one of the supported databases. Review the requirements for database, instance, and user names, and specific parameters for the appropriate database manager in the WebSphere Portal Information Center. Pay attention to the Oracle pre-WebSphere Portal installation setup.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>▶ Yes, but the database is not one of the supported versions or databases.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Have you identified the type, if any, of LDAP that will be used?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>▶ Yes. Review the LDAP installation section of the WebSphere Portal V5.1 Information Center.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>▶ Yes. Verify that the WebSphere Portal LDAP schema will fit into an existing customer LDAP.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>▶ Yes. A custom user registry (CUR) will be used.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table D-8  Solution Assurance Review checklist: Implementation and operation

<table>
<thead>
<tr>
<th>Activity: Implementation and operation</th>
<th>Y/N or N/A</th>
<th>Factors increasing risk</th>
<th>Factors reducing risk</th>
<th>Task owner</th>
<th>Due date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Does a project plan exist for the implementation and has it been produced or reviewed by someone with relevant experience? Has a project manager been included?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Has a post sales support contract been included in the proposal to cover the operational hours?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Is it explicitly stated who is responsible for systems management, both defining the processes and tools and providing the services (change/problem/performance/availability/capacity/operations)?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Are there any formal acceptance tests? Who owns them and has sufficient resource been planned to produce, negotiate, and execute them?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Has a training and education plan been defined and documented?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>What are the plans for a test environment? If the customer has no plans, mark this as a high-risk item in this document and document an action plan and owner to address this.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
What are the plans to test this technical solution? Do the plans incorporate use of:
- A sophisticated load simulation tool
- Creation and use of load scripts
- Creation and execution of testing scenarios
- Adequate knowledge of representative use cases

If the customer/implementer has no plans, mark this as a high-risk item in this document and document an action plan and owner to address this.

Will the solution be stress tested?

Are there plans for final performance tuning of the application server?

Has time for testing been allocated for this in the project plan?

What plans are in place to generate new test scenarios when new code is written during any stage of the application?

<table>
<thead>
<tr>
<th>Activity: Services and education</th>
<th>Y/N</th>
<th>Factors increasing risk</th>
<th>Factors reducing risk</th>
<th>Task owner</th>
<th>Due date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Are there any factors that might prevent the customer from successfully completing the proposed project by themselves?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Is the customer willing to include services as part of the solution?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Who will provide the services if any will be used, and is there a formal commitment to provide the services?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Has the services provider quoted for the services and has this been included in the proposal? Does the service provider have adequate skills?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Where more than one provider is responsible, for example, customer/Business Partner/IBM, is it clear who is responsible for what? Are completion criteria clear?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Does the solution include education that addresses the areas of application development, implementation, as well as production, operations, and administration?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>If education is included, has an appropriate education provider been identified and engaged?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Activity: Skills assessment</td>
<td>Y/N</td>
<td>Factors increasing risk</td>
<td>Factors reducing risk</td>
<td>Task owner</td>
<td>Due date</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-----</td>
<td>-------------------------</td>
<td>-----------------------</td>
<td>------------</td>
<td>----------</td>
</tr>
<tr>
<td>Have at least some project members successfully completed any earlier projects using WebSphere Portal?</td>
<td>Y/N</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>▶ Yes. At least some project members have previous WebSphere Portal V5.1 experience.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>▶ Yes. At least some project members have previous WebSphere Portal V5.0 experience.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>▶ Yes. At least some project members have previous WebSphere Portal V4.1 or V4.2 experience.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Does the architecture of the overall system require complex WebSphere administration (for example, clustering, fail-over management, and performance tuning)?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>What skill level is available for architecting the application server (0-5)? See Table D-17 for IBM Skills Assessment Ratings.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>What skill level is available for administering the application server (0-5)?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>What skill level is available for administering the Web server (0-5)?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Does the project team contain members with appropriate Lotus product skills?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>▶ IBM Workplace Web Content Management</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>▶ IBM Lotus Sametime</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>▶ IBM Lotus QuickPlace</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>▶ Set up and use of Lotus Domino Server</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>▶ LDAP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Are Java programming skills required for this project?  
If so, what skill level is available (0-5)?

Are JSP/servlet development skills required for this project?  
If so, what skill level is available (0-5)?

Are Portlet development and packaging skills required for this project?  
If so, what skill level is available (0-5)?

What level of Portlet administration skills are available (0-5)?

Are EJB development and packaging skills required for this project?  
▷ If so, what skill level is available (0-5)?  
▷ If so, have the transactional capabilities been documented?  
▷ What types of EJBs are in the solution?  
   – Stateful  
   – Stateless  
   – Entity bean managed  
   – Entity container managed

Are custom user registry skills required for this job?  
▷ If so, what skill level is available?  
▷ If so, does the team have experience working with and coding to the interfaces for the WebSphere Application Server CUR and the WebSphere Member Manager Repository?  
▷ Contact the WebSphere Portal Development lab through a PMR if more information is needed on the WebSphere Portal Member Manager Repository requirements.

Are Web service development/deployment skills required for this project?  
If so, what skill level is available (0-5)?

Is performance tuning required for this project?  
If so what skill level is available (0-5)?

What skill level is available for LDAP administration/integration (0-5)?
<table>
<thead>
<tr>
<th>What skill level is available for Netegrity SiteMinder integration (0-5)?</th>
</tr>
</thead>
<tbody>
<tr>
<td>What skill level is available for Tivoli Access Manager integration (0-5)?</td>
</tr>
<tr>
<td>What skill level is available for other third-party authentication and authorization integration (0-5)?</td>
</tr>
<tr>
<td>Are JCA skills required for this project? If so what skill level is available (0-5)?</td>
</tr>
<tr>
<td>Are WebSphere MQ skills required for this project? If so what skill level is available (0-5)?</td>
</tr>
<tr>
<td>What skill level is available for pervasive device support (transcoding) (0-5)?</td>
</tr>
<tr>
<td>Does the project have at least functional admin skills for each operating system running some solution component (portal server, Web server, app server, admin repository, data store, enterprise server)?</td>
</tr>
<tr>
<td>▶ Yes, but not on all operating systems.</td>
</tr>
<tr>
<td>▶ Yes, functional admin skills are available for all necessary operating systems.</td>
</tr>
<tr>
<td>Does the project have at least functional DBA and other required admin skills for each database to be used by the solution (includes admin repository and other application databases)?</td>
</tr>
<tr>
<td>▶ Yes, but not for all databases.</td>
</tr>
<tr>
<td>▶ Yes, functional DBA and admin skills are available for all databases to be used.</td>
</tr>
</tbody>
</table>

Table D-11  Solution Assurance Review checklist: Security assessment

<table>
<thead>
<tr>
<th>Activity: Security assessment</th>
<th>Y/N</th>
<th>Factors increasing risk</th>
<th>Factors reducing risk</th>
<th>Task owner</th>
<th>Due date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Does the solution require separation of the Web server and application server?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Will it be necessary to secure resources on the Web server?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Will further security (such as basic or certificate) other than the form based portal login be required?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Has the appropriate security for your application server been assessed and decided upon?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
What type of user authentication is required for the project?
- Using RDBMS (for example, DB2, Oracle); no LDAP.
- LDAP; state which directory.
- Supported third party; describe.
- Unsupported third party; describe.
- Custom user registry/member repository; describe.

Will the credential service be used and has the mechanism to communicate with enterprise authorization systems been established?

Will the credential service use the credential vault and is the customer satisfied with the security of the vault?

Has the level of SSO implementation been agreed with the customer?

Does the user registration process need to be customized (for example, validated against third-party process, database, transaction system, and so on) and is it understood how this will be accomplished?

Will reverse proxies be used? For example, many customers front their portal server with WebSEAL. WebSEAL can function both as a reverse proxy and also has a Tivoli Access Manager plug-in such that it will communicate directly with Tivoli Access Manager for authentication and authorization.

If Tivoli Access Manager will be used, have you read the guidance at: http://publib.boulder.ibm.com/pvc/wp/510/ent/en/InfoCenter/wpf/conf_pdr.html

Will the customer use self-registration or make use of a customer service rep to enroll users?

<table>
<thead>
<tr>
<th>Activity: Performance assessment</th>
<th>Y/N</th>
<th>Factors increasing risk</th>
<th>Factors reducing risk</th>
<th>Task owner</th>
<th>Due date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Has the documentation section about optimizing performance been read?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Has the Capacity Planning Guide for WebSphere Portal 5.1, WebSphere 5.x, the chosen database, and LDAP server been read? Review the Performance Tuning Guide for WebSphere Portal V5.1 at: <a href="http://www.ibm.com/support/search.wss?rs=688&amp;tc=S5HRXX&amp;q=tuning">http://www.ibm.com/support/search.wss?rs=688&amp;tc=S5HRXX&amp;q=tuning</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Have you considered caching and edge of network servers?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table D-13  Solution Assurance Review checklist: Development environment

<table>
<thead>
<tr>
<th>Activity: Development environment</th>
<th>Y/N</th>
<th>Factors increasing risk</th>
<th>Factors reducing risk</th>
<th>Task owner</th>
<th>Due date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Which integrated development environment (IDE) is to be used for the project?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>➤ Rational Application Developer.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>➤ Rational Web Developer.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>➤ J Builder.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>➤ Sun IDE.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>➤ Text Editor and JDK™ tools.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>➤ Other; describe.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Does your chosen development environment provide integrated source level debugger, persistence mapping to legacy data, and easy deployment?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Does your chosen development environment provide integrated source level debugger and test environment for portlets and easy deployment of portlets?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table D-14  Solution Assurance Review checklist: Legacy data requirements

<table>
<thead>
<tr>
<th>Activity: Legacy data requirements</th>
<th>Y/N</th>
<th>Factors increasing risk</th>
<th>Factors reducing risk</th>
<th>Task owner</th>
<th>Due date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Which of the following enterprise servers are to be accessed by the application? Indicate skill level (0-5) for administering each system that will be accessed (see Table D-17).</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>➤ PeopleSoft.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>➤ Oracle.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>➤ WebSphere MQ.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>➤ Third-party Web content management (state which server).</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>➤ SAP.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>➤ Baan.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>➤ Siebel.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>➤ J.D. Edwards.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>➤ Other; describe.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Will this application include model business processes that span multiple back ends?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Is two-phase commit a requirement of any of the business processes being modeled?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Activity: Hardware configuration</td>
<td>Y/N NA</td>
<td>Factors increasing risk</td>
<td>Factors reducing risk</td>
<td>Task owner</td>
<td>Due date</td>
</tr>
<tr>
<td>-------------------------------------------------------------------------------------------------</td>
<td>--------</td>
<td>-------------------------</td>
<td>-----------------------</td>
<td>------------</td>
<td>---------</td>
</tr>
<tr>
<td>Has the hardware configuration been sized and verified to match the customer's requirements?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Have the following item been taken into consideration?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>▶ Cost.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>▶ Number of users, applications, nodes, partitions.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>▶ Footprint (physical size of the machines).</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>▶ Scalability and high availability.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>▶ Does each machine meet the component hardware requirement?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>▶ How many machines will be required?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>▶ Redundancy required for each machine?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Does the machine have multiple LPARs? This adds another layer of complexity.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Have hardware requirements for other software needed to implement the full solution been documented and verified?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Does the fully configured machine allow for expected growth (vertical scalability)?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Will the planned configuration provide the required throughput and response time?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Could the configuration be expanded easily to meet future growth?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Has the customer planned client, server, and host connectivity options?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Are the availability requirements known and documented and does the proposed configuration deliver this?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Are hardware maintenance requirements known and agreed?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Have components been identified for the customer's systems backup and recovery procedures?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Have system management tools and services been proposed?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Have required system management consoles been configured?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table D-16  Solution Assurance Review checklist: Communications and networking

<table>
<thead>
<tr>
<th>Activity: Communications and networking</th>
<th>Y/N</th>
<th>Factors increasing risk</th>
<th>Factors reducing risk</th>
<th>Task owner</th>
<th>Due date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is the appropriate network infrastructure in place (LAN, WAN, router, bridge, cabling)?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Are network schematics documented and available?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Have the appropriate connectivity options been explored and decided upon?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Has the communication protocol been decided upon?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Are additional networking hardware devices (routers, switches, and so on) required to implement the solution?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>If pervasive device support is required, are the appropriate gateways or service providers in place and understood for accessing the secured network?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Will a Domain Name System (DNS) be employed in the network?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>If firewalls are employed, will specific ports be able to be opened?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: Is anyone aware of any risks or issues that have not been adequately explored?
Is there any other information that would be useful for the SA team members to be aware?

In Table D-17, all skill levels refer to IBM Skills Assessment Ratings.

Table D-17  IBM Skills Assessment Ratings

<table>
<thead>
<tr>
<th>Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level 5</td>
<td>Has comprehensive knowledge with ability to make sound judgments. Can give expert advice and lead others to perform. Extensive and comprehensive experience.</td>
</tr>
<tr>
<td>Level 4</td>
<td>Has in-depth knowledge and can perform without assistance. Can direct others in performing. Repeated and successful experience.</td>
</tr>
<tr>
<td>Level 3</td>
<td>Can perform with assistance. Has applied knowledge. Has performed with assistance on multiple occasions. Has performed independently in routine situations.</td>
</tr>
<tr>
<td>Level 2</td>
<td>Limited ability to perform. Has general knowledge only. Very limited experience.</td>
</tr>
<tr>
<td>Level 1</td>
<td>Limited knowledge. No experience.</td>
</tr>
<tr>
<td>Level 0</td>
<td>No knowledge. No experience.</td>
</tr>
</tbody>
</table>
Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this Redpaper.

IBM Redbooks

For information about ordering these publications, see “How to get IBM Redbooks” on page 136. Note that some of the documents referenced here may be available in softcopy only.

- IBM WebSphere Everyplace Access V5 Handbook for Developers and Administrators
  Volume I: Installation and Administration, SG24-6462
- IBM WebSphere Everyplace Access V5 Handbook for Developers and Administrators
  Volume II: Application Development, SG24-6463
- IBM WebSphere Everyplace Access V5 Handbook for Developers and Administrators
  Volume III: E-Mail and Database Synchronization, SG24-6676
- IBM WebSphere Everyplace Access V5 Handbook for Developers and Administrators
  Volume IV: Advanced Topics, SG24-6677
- Architecting Portal Solutions, SG24-7011
- Develop and Deploy a Secure Portal Solution Using WebSphere Portal V5 and Tivoli
  Access Manager V5.1, SG24-6325
- IBM Rational Application Developer V6 Portlet Application Development and Portal Tools,
  SG24-6681
- IBM Workplace Web Content Management for Portal 5.1 and IBM Workplace Web
  Content Management 2.5, SG24-6792
- IBM WebSphere V5.1 Performance, Scalability, and High Availability WebSphere
  Handbook Series, SG24-6198

Other publications

These publications are also relevant as further information sources:

  0471453803

Online resources

These Web sites and URLs are also relevant as further information sources:

- IBM Support
  http://www.ibm.com/support
IBM Software Support
http://www.ibm.com/software/support/

IBM software support problem submission page

IBM Software Support Handbook
http://techsupport.services.ibm.com/guides/handbook.html

IBM developerWorks
http://www.ibm.com/developerworks/

IBM developerWorks WebSphere Portal zone

IBM Passport Advantage
http://www.ibm.com/software/passportadvantage

WebSphere Portal for Multiplatforms Information Center
http://publib.boulder.ibm.com/infocenter/wpdoc/v510/index.jsp

IBM Workplace Solutions Catalog
http://catalog.lotus.com/wps/portal/workplace

IBM Workplace Solutions Catalog: WebSphere Portal
http://catalog.lotus.com/wps/portal/portal

WebSphere Edge Server

WebSphere Edge Server documentation

WebSphere Portal Automated Problem Determination Tool Version 1.1.4

Service Oriented Architecture (SOA)
http://www.ibm.com/SOA

Business Value Assessment for Portal
http://www.ibm.com/industries/travel/doc/content/bin/050622_Business_Value_Assesment_for _Workplaces_brochure.pdf

Return on investment study

IBM Workplace Web Content Management
http://www.ibm.com/software/workplace/webcontentmanagement

WebSphere Application Server Information Center
http://www.ibm.com/software/webservers/appserv/was/library/

Rational software Web site

Rational ClearCase

Rational Suite
- Tivoli Web Site Analyzer
- IBM Tivoli Composite Application Monitor for WebSphere
- IBM Tivoli Composite Application Monitor for Response Time Tracking
- Lotus Documentation
  http://www.lotus.com/ldd/notesua.nsf/find/st30
- IBM Workplace Web Content Management Version 2.5 - Installation Guide
- SurfAid
- IBM WebSphere Portal Version 5.1 Tuning Guide
  http://www.ibm.com/support/docview.wss?uid=swg27005771&aid=1
- Integrating WebSphere Portal software with your security infrastructure white paper
- Using WebSphere Dynamic Cache Service with IBM Workplace Web Content Management
- Creating portal instances on demand
- Setting up Portal Search in a WebSphere Portal V5.1 clustered environment
- A step-by-step guide to configuring a WebSphere Portal V5.1 cluster using WebSphere Application Server V5.1.1.1
- Continuous availability maintenance procedures for the IBM WebSphere Portal 5.1.x environment
- Portlet API Comparison White Paper: JSR 168 Java Portlet Specification compared to the IBM Portlet API
- Developing JSF Portlets with Rational Application Developer 6.0 and WebSphere Portal Server 5.1
  http://www.ibm.com/developerworks/rational/library/05/genkin/
- Examples of themes and skins
How to use WebSphere Everyplace Mobile Portal

Tivoli Access Manager technote

Sametime and Domino portlets

Single Sign-on in a Multi-directory World: “Never say login again”

Develop high performance Web sites with both static and dynamic content using WebSphere Portal 5.1

Caching data in JSR 168 portlets with WebSphere Portal 5.1

Enhancing portal page rendering performance by managing long-running back-end calls, which includes information about the use of the dynamic cache distributed map

Creating and deploying a portlet service for IBM and JSR 168 portlets, a tutorial that includes information about command caches

A Guide to using WebSphere Portal Search- First steps

Using the command cache to improve portal application performance

Static and dynamic caching in WebSphere Application Server V5, a description not specifically for WebSphere Portal but with some general information about caching that is also true for portal systems

Comparing the JSR 168 Java Portlet Specification with the IBM Portlet API

Bill Hines: My (least) favorite anti-practices

“Awareness, Connection and Authentication Problems if Collaborative Portlets v5.1 not Configured for Tivoli Access Manager” technote

Portal maintenance
- **Java Diagnostics Guide**
- Wikipedia Reliable system design entry (single point of failure)
  http://en.wikipedia.org/wiki/Reliable_system_design
- Wikipedia Service Level Agreement (SLA) entry
  http://en.wikipedia.org/wiki/Service_Level_Agreement
- Wikipedia Response time entry
  http://en.wikipedia.org/wiki/Response_time
- Wikipedia Security entry
- OASIS Web Services for Remote Portlets (WSRP)
- wget, a non-interactive command line tool
  http://www.gnu.org/software/wget/wget.html
- Ethereal
  http://www.ethereal.com
- VeriSign
  http://www.verisign.com
- Thawte
  http://www.thawte.com
- Microsoft Internet Explorer Renegotiates Secure Sockets Layer Connection Every Two Minutes article
  http://support.microsoft.com/kb/q265369/
- HTML TIDY
  http://www.w3.org/People/Ragget/tidy/
- Coley Consulting, Why Projects Fail
  http://www.coleyconsulting.co.uk/failure.htm
- Extreme Programming
  http://www.extremeprogramming.org
- Concurrent Versions System (CVS)
  http://www.nongnu.org/cvs/
- VMware
  http://www.vmware.com
- Mercury LoadRunner
- Open Source tool Bugzilla
  http://www.bugzilla.org
- Wilytech Introscope
- Ascera Manager 5 for Portal
  http://www.acsera.com/section/view/wces/88/
How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services
This IBM Redpaper is designed to provide a road map of information about how to best plan and ensure a successful deployment of IBM WebSphere Portal into an organization.

In this Redpaper, we provide you with an understanding of the WebSphere Portal technology and discuss common goals for customer portal projects. In addition, we discuss initial planning considerations and how to plan your architecture for deployment.

Best practices for a successful portal deployment requires a systematic process. Therefore, we address the initial questions that are asked to get you started and keep you on the right path. You will find that this best practices guide also serves as your systems assurance lead, ensuring that IBM products are planned, shipped, installed, and tested in such a way that clients derive maximum satisfaction with minimal disruption.

This document is designed for the following audience: IBM employees, IBM Business Partners, and clients.